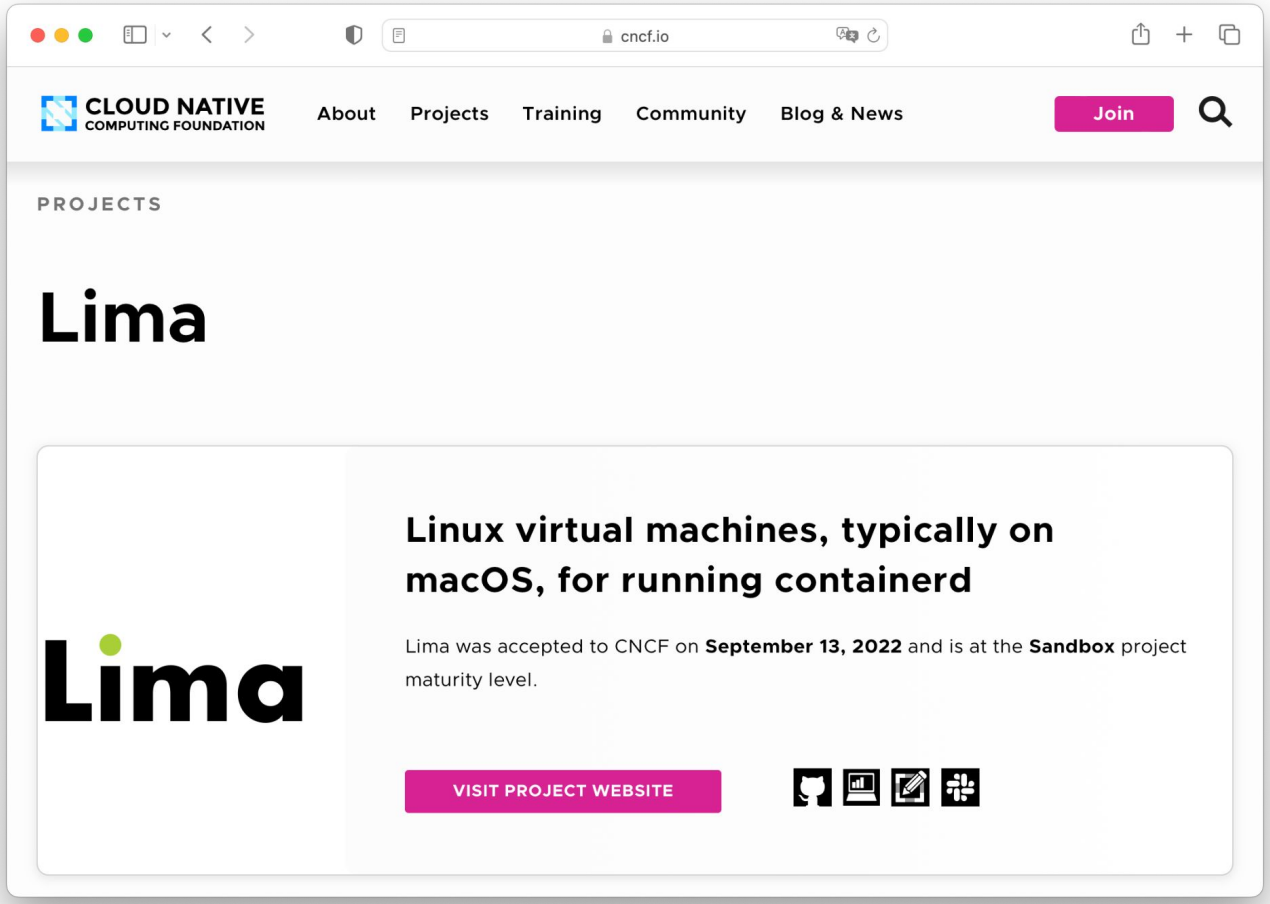

**Linux virtual machines, typically on macOS, for running containerd**
**https://github.com/lima-vm/lima**

Akihiro Suda, NTT



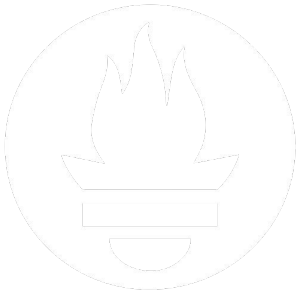
Relevant talk at KubeCon EU (May 2022): "Running containerd and k3s on macOS" - Akihiro Suda, NTT Corporation & Jan Dubois, SUSE

# Lima joined CNCF Sandbox




https://www.cncf.io/projects/lima/

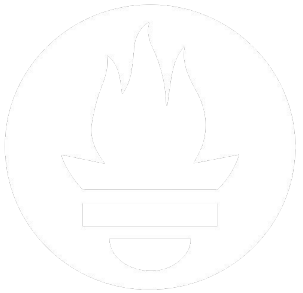
# Why run containers on macOS?




- 2022 is *The Year of the Linux Desktop*™…
- But ordinary developers still need macOS (or Windows)

- Almost solely for the dev & test environment
- Not the best fit for running a production server

# Existing methods

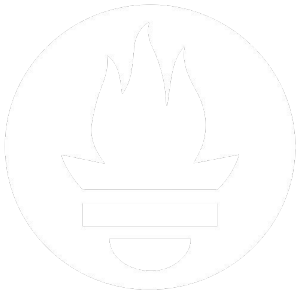




- Docker Desktop for Mac has been the popular solution

- Supports automatic host filesystem sharing
- Supports automatic port forwarding
- But proprietary

# Existing methods



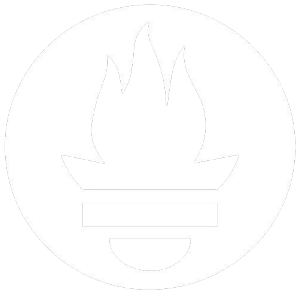


Just install Docker and Kubernetes inside a Linux VM?

Maybe via minikube?

- VMware Fusion and Parallels are proprietary
- VirtualBox is FLOSS but won't support M1
- QEMU is FLOSS and supports M1, but still
  - Not easy to access the host FS from the containers
  - Not easy to access the container ports from the host

# Our solution: Lima





- Similar to WSL2 but for macOS hosts

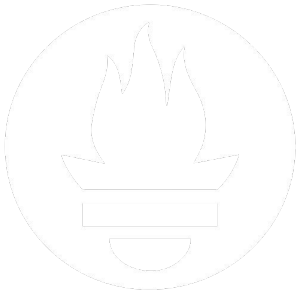
- Automatic host filesystem sharing
- Automatic port forwarding
- Built-in integration for containerd

```
$ brew install lima
$ limactl start
$ lima nerdctl run ...
```

Lima

https://github.com/lima-vm/lima

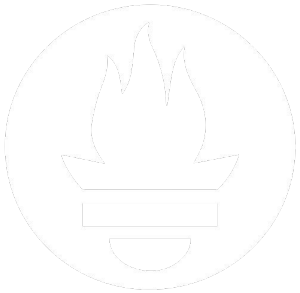
# Lima = LInux MAchine





- Originally designed as “*containerd machine*” to mimic Docker Machine
- The scope was extended immediately to cover other use cases too
- Still focuses on containerd and k3s

# containerd with Lima



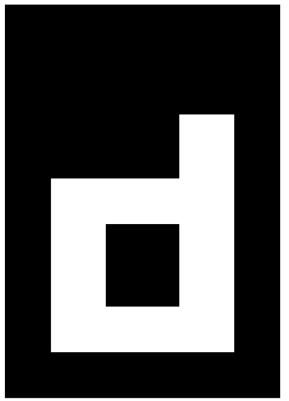


**containerd**: the de facto standard container runtime

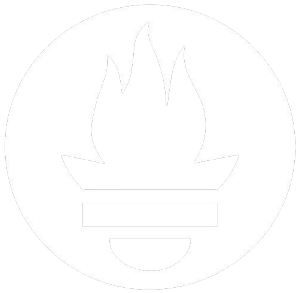
- CNCF Graduated project
- Not just made for Kubernetes
- Provides the docker-compatible CLI too: **nerdctl**

```
$ nerdctl build -t foo .
$ nerdctl run -d -p 127.0.0.1:80:80 foo
```

- With a lot of cutting-edge features
  - Lazy-pulling, IPFS, OCIcrypt, Faster rootless …



# containerd with Lima





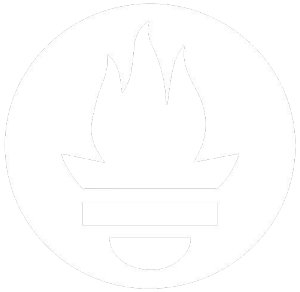
Lima provides built-in support for containerd

Build an image from a Dockerfile on the macOS home directory

```
$ lima nerdctl build -t foo .
$ lima nerdctl run -d -p 127.0.0.1:80:80 foo
```

Expose the container's port 80 as the macOS's http://localhost

# containerd with Lima



Even supports running Intel (AMD64) containers on M1/M2 (ARM64) and vice versa, using tonistiigi/binfmt



Run an AMD64 container on M1/M2 (ARM64)

```
$ lima nerdctl run --platform=amd64 ...
```

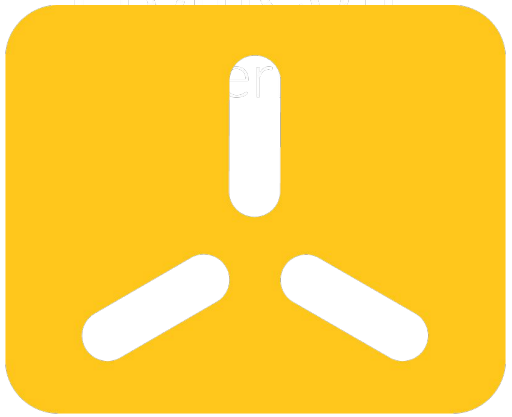
Build an AMD64/ARM64 dual-platform image

```
$ lima nerdctl build --platform=amd64,arm64 ...
```

# k3s with Lima

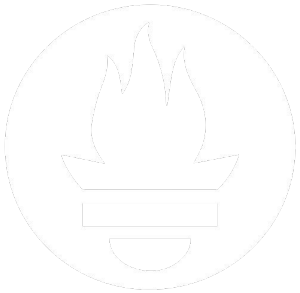
**k3s**: Lightweight Kubernetes

- CNCF Sandbox project
- Adopts containerd as the CRI runtime
- Works with Lima too



```
$ limactl start template://k3s
$ limactl shell k3s sudo cat /etc/rancher/k3s/k3s.yaml \
  >~/.kube/config
$ kubectl ...
```

# Extra: Docker with Lima



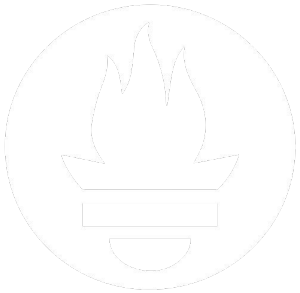


The original design was only to support containerd, but the scope is now expanded to support Docker Engine too

(Docker Engine: Apache License 2.0, no proprietary GUI)

```
$ limactl start template://docker
$ brew install docker
$ docker context create lima --docker \
  "host=unix://$HOME/.lima/docker/sock/docker.sock"
$ docker context use lima
$ docker run ...
```

# Extra: Podman with Lima

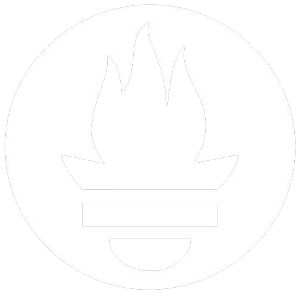


PromCon
North America 2021

And even Podman

```
$ limactl start template://podman
$ brew install podman
$ podman system connection add lima \
  "unix://$HOME/.lima/podman/sock/podman.sock"
$ podman system connection default lima
$ podman run ...
```

# How it works: Hypervisor

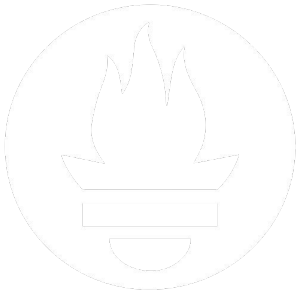




- Vanilla QEMU
- Supports both Intel and ARM
- Even supports Intel-on-ARM and ARM-on-Intel (slow though)

- **FAQ**: why not use Apple's Virtualization.framework?
  - Proprietary
  - Limited functionalities

# How it works: Filesystem sharing

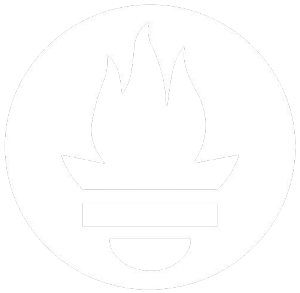




- **Lima < 1.0**: reverse SSHFS
  - macOS works as an SSH client but as an SFTP server
  - Linux works an SSH server but as an SFTP client

- **Lima ≥ 1.0**: virtio-9p-pci , aka virtfs (not virtio-fs)
  - Less weirdness, tolerant of Ethernet failure
  - Lima 1.0 will be released by the end of the year

# How it works: Filesystem sharing





- **FAQ**: why not use virtio-fs (faster than virtfs) ?
  - QEMU still doesn’t implement virtio-fs for macOS hosts
  - Apple’s Virtualization.framework implements virtio-fs, but it is proprietary and lacks other functionalities

# How it works: Port forwarding

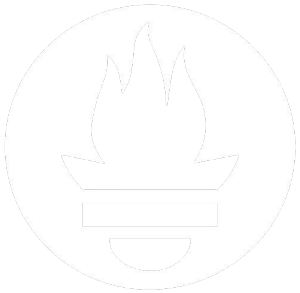


- The guest ports are accessible as localhost from the host

- Watch guest events, and run `ssh -L` to let SSH forward TCP ports

- Event sources:
  - `/proc/net/{tcp,tcp6}`: For non-CNI ports
  - `iptables`, `AUDIT_NETFILTER_CFG`: For CNI ports

# How it works: Networking

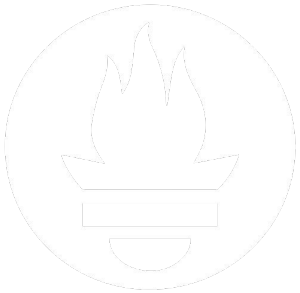




The default networking is QEMU's -netdev user (aka slirp)

- No root privilege is needed at all
- The guest IP is not reachable from the host and other VMs
  (But Lima forwards all localhost ports)
- Especially problematic for multi-node Kubernetes

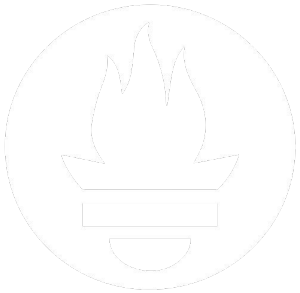
# How it works: Networking




**Opt-in**: `socket_vmnet` (https://github.com/lima-vm/socket_vmnet)

- Assign “real” IP reachable from the host, other VMs, and even from other hosts (with bridge mode)
- Caveat: root privilege is needed for running `socket_vmnet` daemon (not for QEMU)

# How it works: Networking



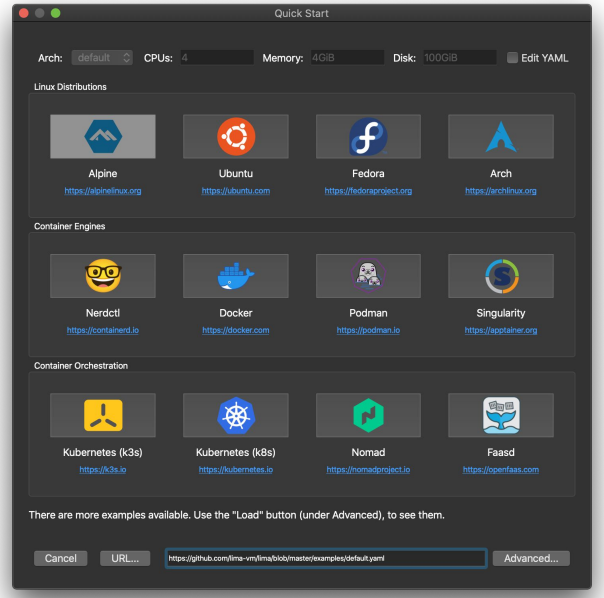


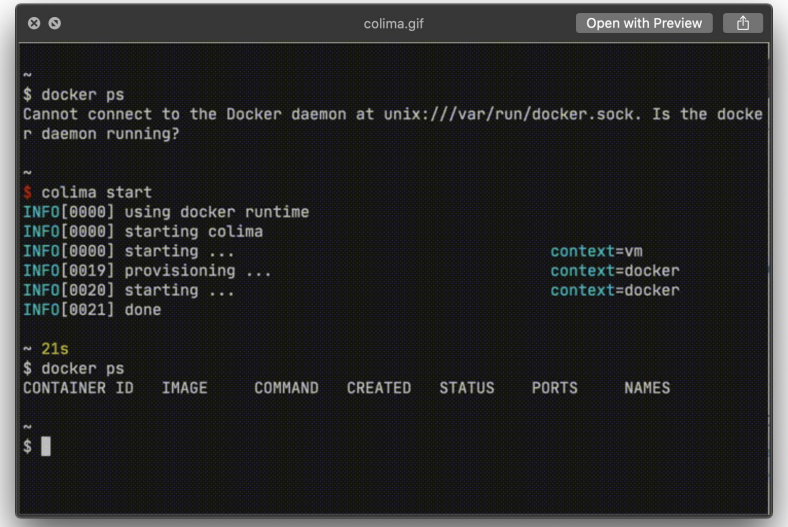
**FAQ**: why not use QEMU's -netdev vmnet-shared ?
(available since QEMU 7.1)

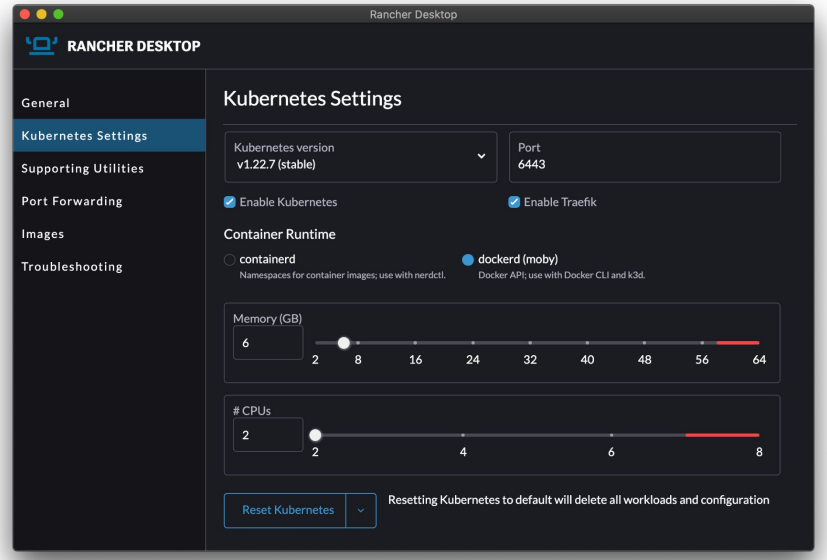
- Because it needs running the entire QEMU as the root

# Third party FOSS projects

| | |
|---|---|
| **Lima-GUI** | https://github.com/afbjorklund/lima-gui |
| **Colima** | https://github.com/abiosoft/colima |
| **Rancher Desktop** | https://github.com/rancher-sandbox/rancher-desktop |

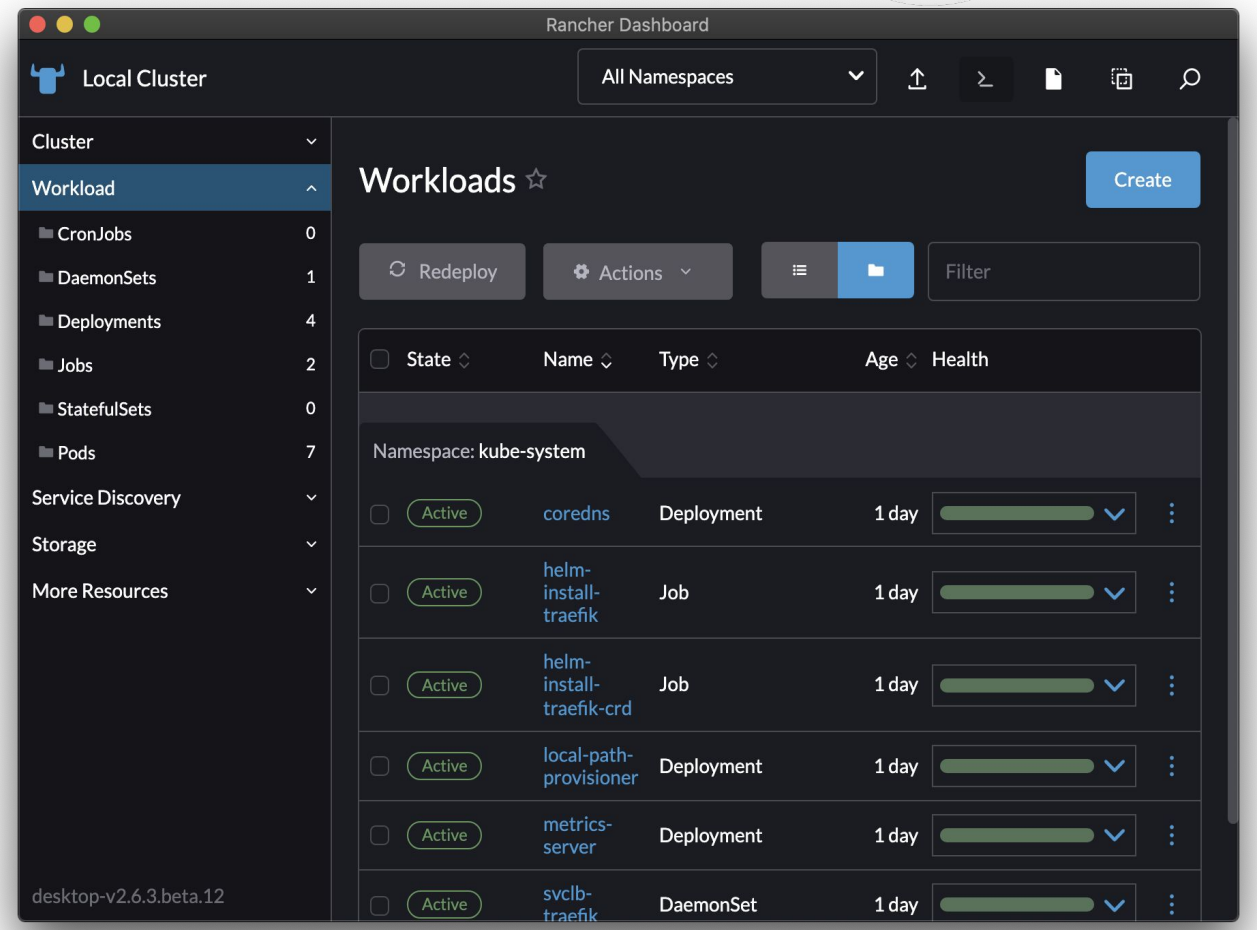


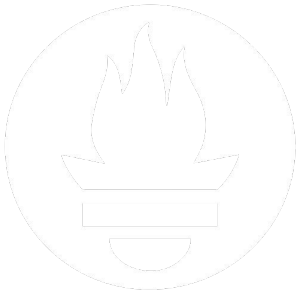



# Rancher Desktop

- GUI for containerd, moby, and k3s
- Rancher Dashboard for Kubernetes
- Test Kubernetes version upgrades
- Image scanning with Trivy
- Also works on Linux & Windows (WSL2)
- Free and open source



# Recap



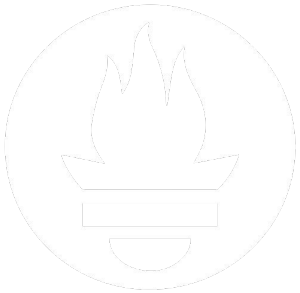


Lima provides a quick way to run containerd and k3s on macOS

- With automatic host filesystem sharing
- With automatic port forwarding

```
$ brew install lima
$ limactl start
$ lima nerdctl run -d -p 127.0.0.1:80:80 nginx:alpine
$ curl http://localhost
```

# Join us!



PromCon
North America 2021

- GitHub Discussions: https://github.com/lima-vm/lima/discussions
- CNCF Slack: **#lima** channel

Lima

https://github.com/lima-vm/lima