KubeCon CloudNativeCon Europe 2022

Running containerd and k3s on macOS

Akihiro Suda, NTT
Jan Dubois, SUSE

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

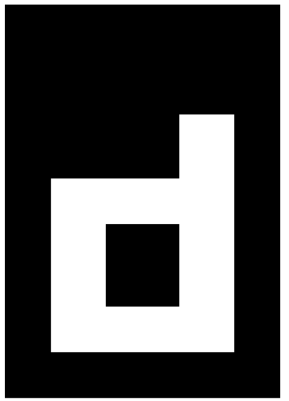
- Originally designed as “*containerd machine*” to mimic Docker Machine
- The scope was extended immediately to cover other use cases too
- Still focuses on containerd and k3s

containerd with Lima

containerd: the de facto standard container runtime

- CNCF Graduated project
- Not just made for Kubernetes
- Provides the docker-compatible CLI too: **contai**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 (ARM64) and vice versa, using tonistiigi/binfmt

Run an AMD64 container on M1 (ARM64)

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

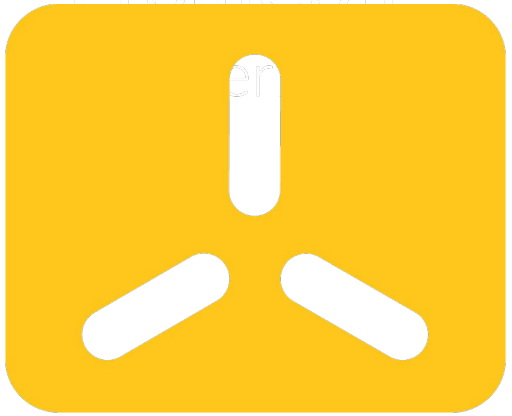
Build an AMD64/ARM64 dual-platform image

```
$ 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

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
  - Lima 1.0 is *probably* available by the time of KubeCon

    (This session was pre-recorded in April)

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 is 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

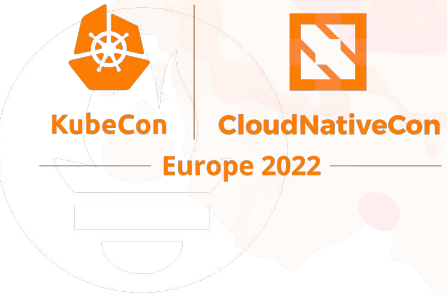
The speaker switches here

KubeCon | CloudNativeCon

Europe 2022

WELCOME TO VALENCIA

Enterprise DNS Requirements

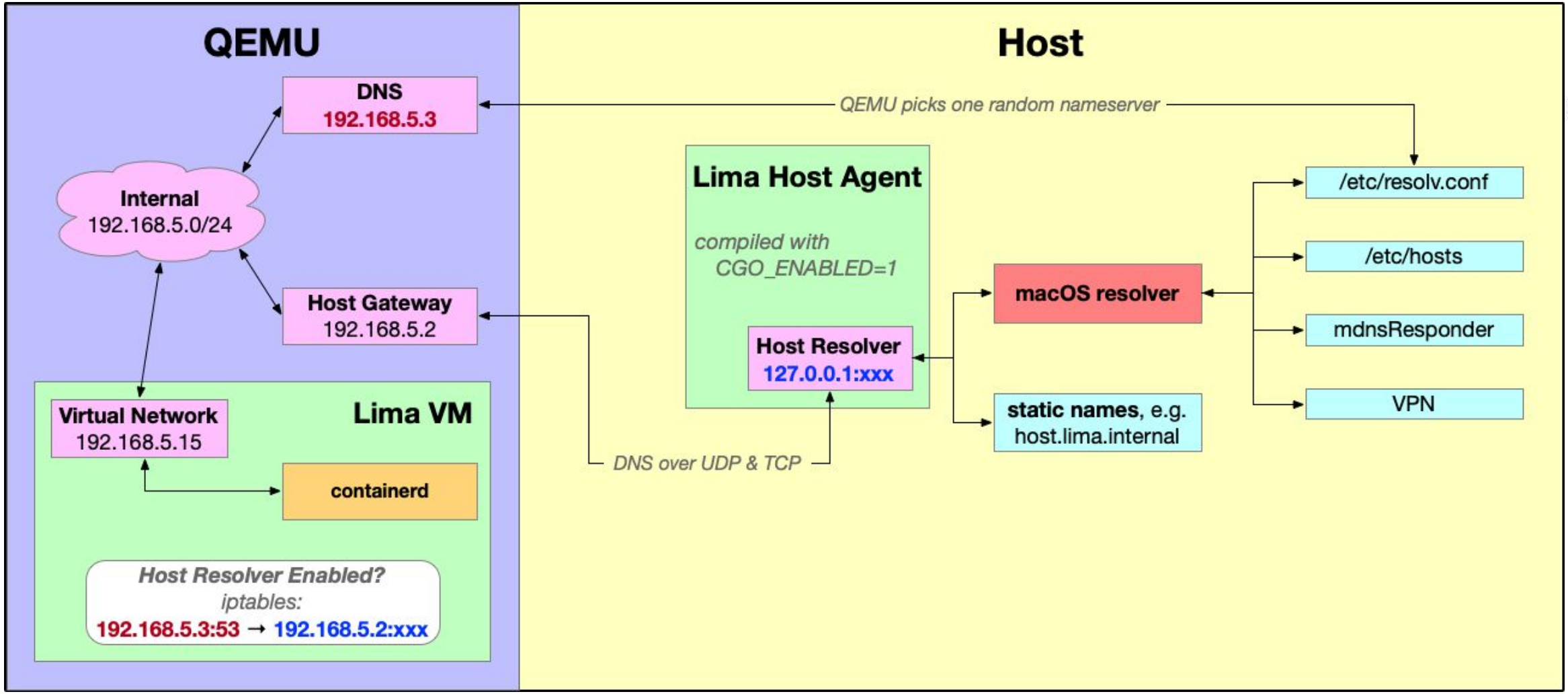


- Use nameservers from VPN connections
- Support for split-DNS

Other QEMU DNS limitations

- Picks single random nameserver from `/etc/resolv.conf`
- Cannot support mDNS
- Doesn’t load `/etc/hosts` from the host

How it works: Host Resolver



How it works: Proxy Settings

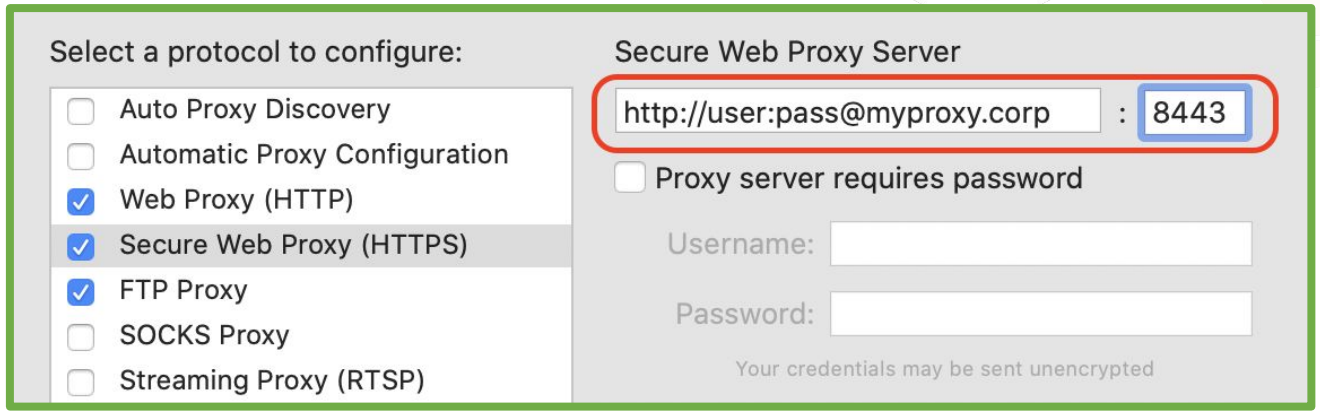
① Network settings

② `lima.yaml`

③ Environment variable

- Change `127.0.0.1` to `192.168.5.2`
- Create matching uppercase and lowercase variants
- Store in `/etc/environment`

①



②

```
File: lima.yaml
env:
  https_proxy: http://127.0.0.1:8888
  http_proxy: http://127.0.0.1:8888
```

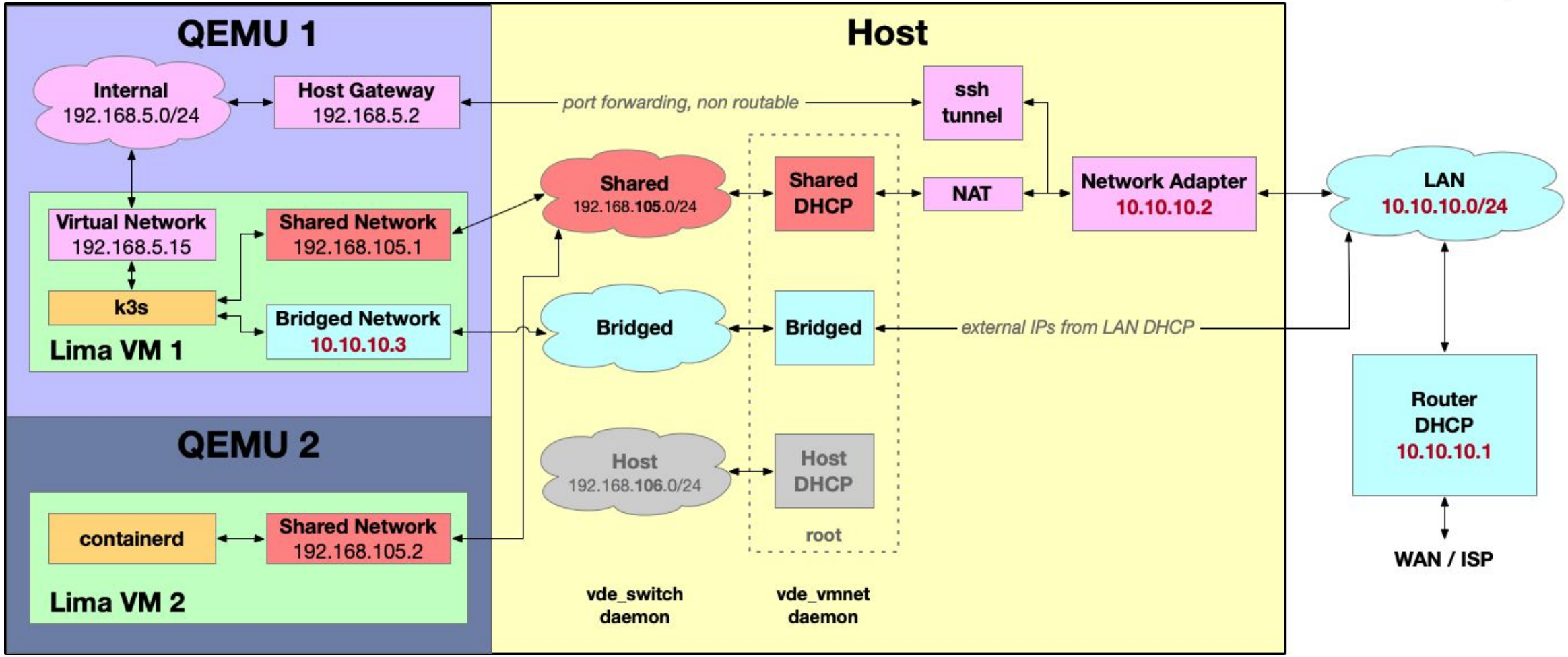
③

```
$ env | grep -i proxy
HTTPS_PROXY=http://proxy.office.com:8080
$ limactl start
```

Port Forwarding Limitations

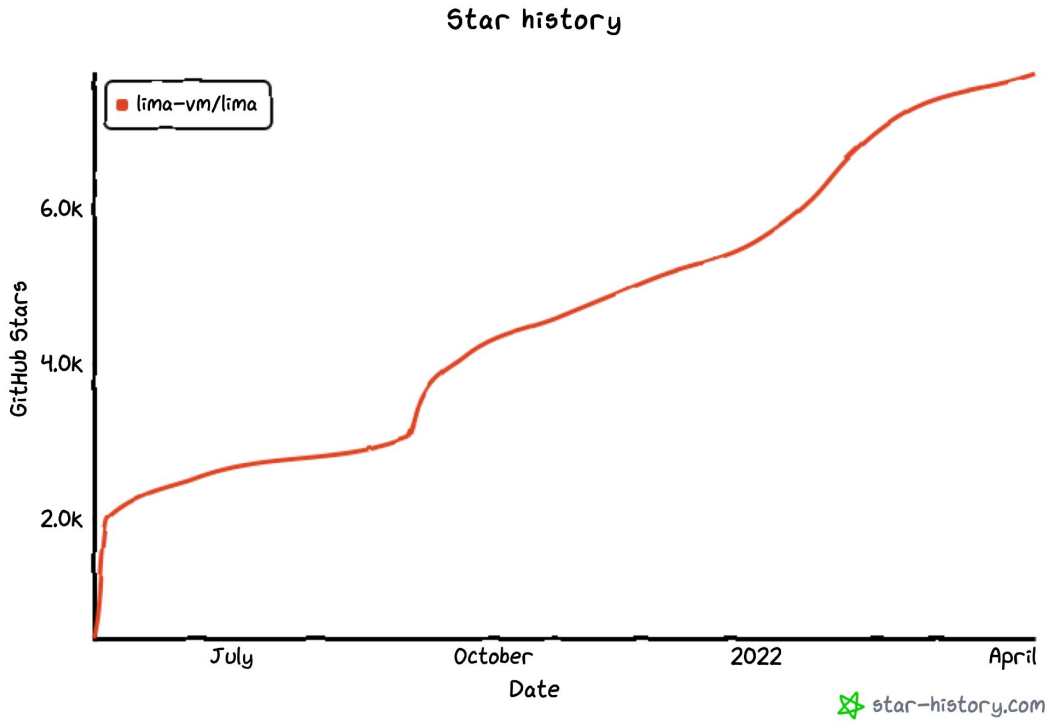
- Port forwarding is delayed up to 3s due to polling
- Port may already be in use on the host
- Guest IP ≠ Host IP breaks external IP for k8s services
- UDP is not supported by ssh port forwarding

How it works: vde_vmnet



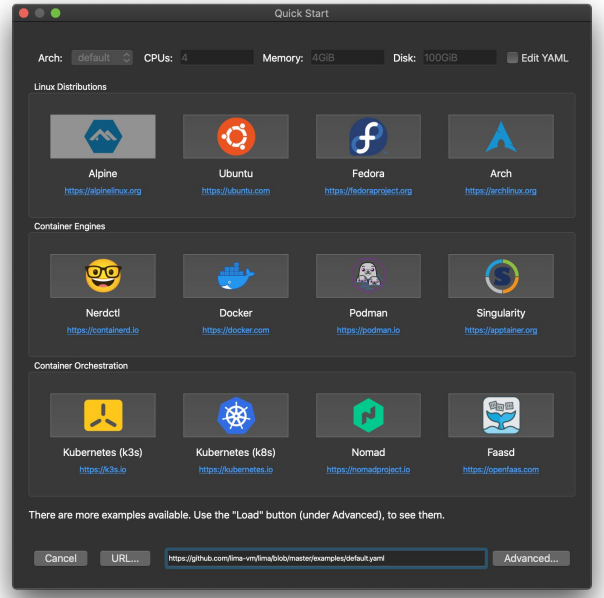
Lima community after one year

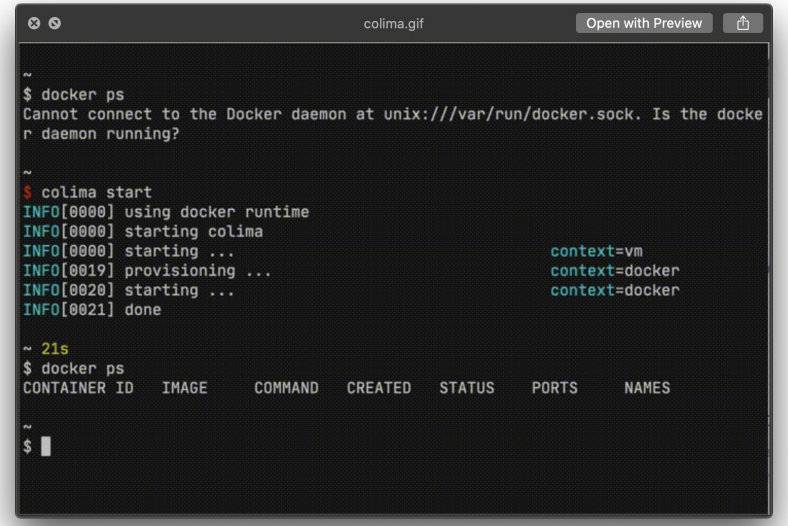
- 45 contributors
- 400 merged pull requests
- 26 releases
- 8k stars on GitHub

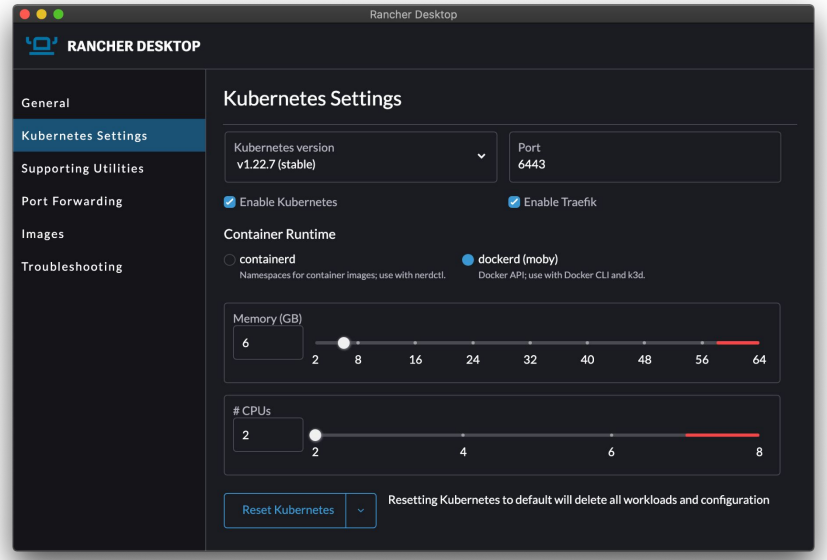


Third party FOSS projects

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

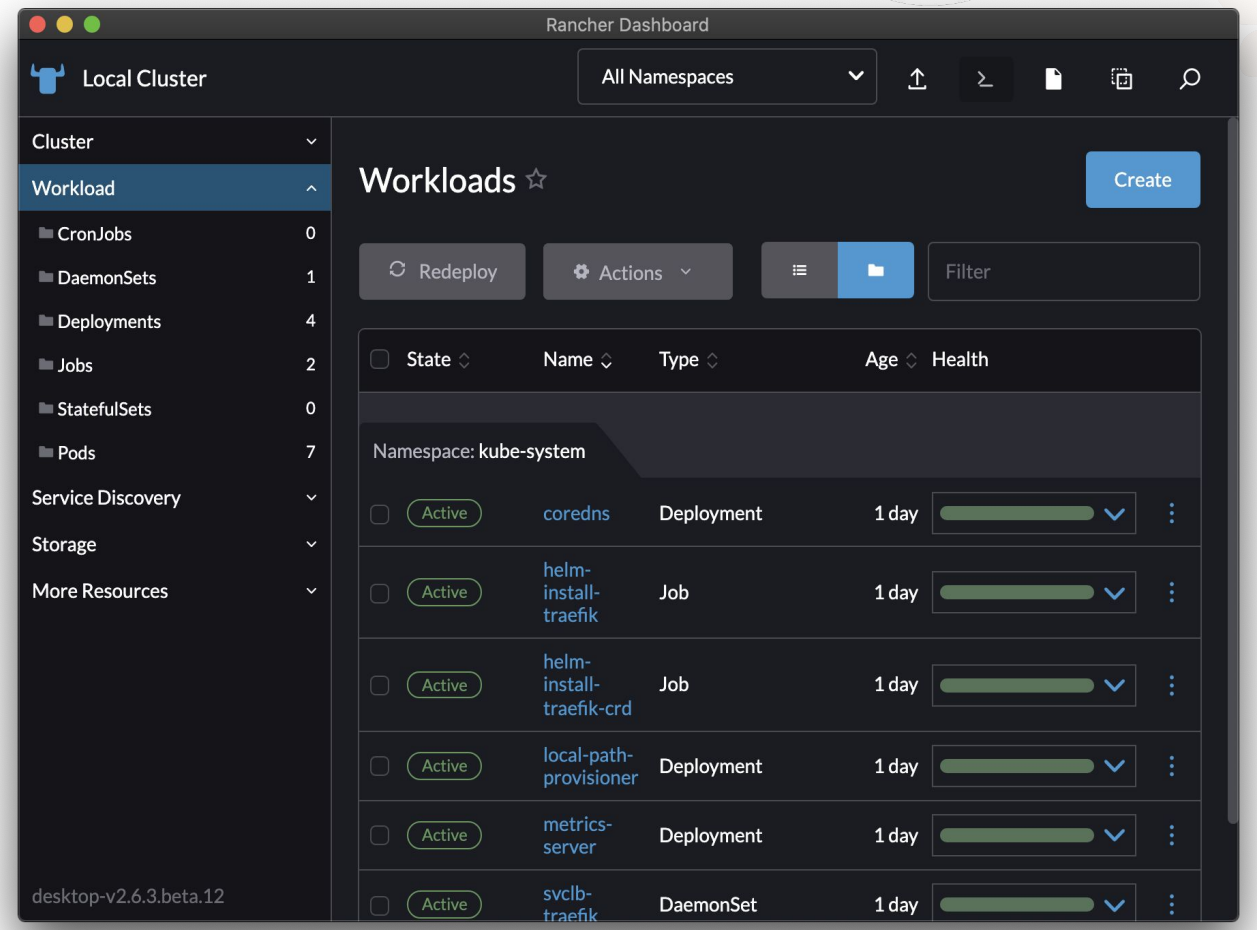






Rancher Desktop

- 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!

- GitHub Discussions: https://github.com/lima-vm/lima/discussions

- Slack: **#lima** channel at https://slack.rancher.io/
  Colocated with #rancher-desktop
  (Lima is not a Rancher project)

Lima

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

KubeCon | CloudNativeCon

Europe 2022

WELCOME TO VALENCIA