

Running Docker (and more) in NetBSD via Lima

Leonardo Taccari

<leot@NetBSD.org>

The NetBSD Foundation

EuroBSDCon 2025 - NetBSD Summit

September 26th 2025

Zagreb, Croatia



Outline

Lima

Linux Machines

How it works?

Defining instances: templates

Instance lifecycle

Demo

Writing a Docker template based on Alpine

Creating instance: `limactl create`

Running the instance: `limactl start`

Using Docker (and more)

Stopping the instance: `limactl stop`

Deleting the instance: `limactl delete`

Status, possible TODOs, help needed

Conclusions

References

Lima

Lima: Linux Machines

- ▶ Lima stands for **L**inux **M**achines
- ▶ Original goal was to promote containerd to Mac users
- ▶ Deploy virtual machines with automatic file sharing and port forwarding
- ▶ Supports several Linux distributions as *guests*
- ▶ Supports several *hosts*: macOS, Linux, NetBSD and DragonFly BSD
- ▶ Daemonless tool: `limactl`



Lima: How it works? ¹

Virtualization QEMU

Networking user-mode emulation (AKA slirp)

Filesystem mounts QEMU virtfs and 9P

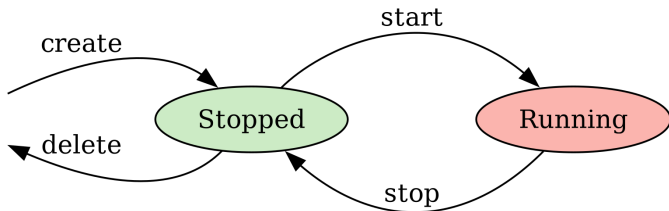
Host and guest agents Permit several communications between host and guest (e.g. automatic port forwarding)

¹We will focus only on Lima on NetBSD and main features/defaults from now on. Please consult the official Lima documentation for more.

Lima: Defining instances: templates

- ▶ Guests are defined via templates
- ▶ Templates are written in YAML
- ▶ Permit to set CPUs, memory, disk, provision scripts and more
- ▶ Lima provides dozens of templates for several distros and versions of them (e.g. Alpine, Debian, Ubuntu, Fedora, etc.)
- ▶ Templates can be customized
- ▶ Every instance is created based on a template

Lima: Instance lifecycle



Lifecycle of Lima instances and status transactions via `limactl` commands.

Demo

Demo: Writing a Docker template based on Alpine I

```
1  minimumLimaVersion: 1.1.0
2
3  base:
4    - template://alpine
5
6  containerd:
7    system: false
8    user: false
9
10 provision:
11 - mode: system
12   script: |
13     #!/bin/sh
14     sed -i 's/host.lima.internal.*/host.lima.internal host.docker.internal/' /
15         etc/hosts
16 - mode: system
17   script: |
18     #!/bin/sh
19     set -eux -o pipefail
20     command -v docker >/dev/null 2>&1 && exit 0
21     apk update
22     apk add docker
23     rc-update add docker default
24     service docker start
25 - mode: user
26   script: |
27     #!/bin/sh
28     set -eux -o pipefail
29     sudo addgroup "$USER" docker
30 probes:
```

Demo: Writing a Docker template based on Alpine II

```
30 - script: |
31     #!/bin/sh
32     set -eux -o pipefail
33     if ! timeout 30s bash -c "until command -v docker >/dev/null 2>&1; do sleep
34         3; done"; then
35         echo >&2 "docker is not installed yet"
36         exit 1
37     fi
38 hostResolver:
39     hosts:
40         host.docker.internal: host.lima.internal
41 portForwards:
42 - guestSocket: "/var/run/docker.sock"
43   hostSocket: "{{.Dir}}/sock/docker.sock"
44 message: |
45     To run 'docker' on the host (assumes docker-cli is installed), run the
46     following commands:
47     -----
48     docker context create lima --docker "host=unix://{{.Dir}}/sock/docker.sock"
49     docker context use lima
50     docker run hello-world
51     -----
```

Demo: Creating instance: `limactl create l`

- ▶ To create the instance we run:

```
$ limactl create \  
  --name=docker \  
  --cpus=2 \  
  --memory=4 \  
  --disk=20 \  
  .../alpine-docker.yaml
```

- ▶ This will fetch the image and populate the instance directory
 `~/.lima/docker`

Demo: Creating instance: `limactl create ll`

Demo: Creating instance: `limactl create (MP4)`

Demo: Running the instance: `limactl start l`

- ▶ To start the instance we run:
`$ limactl start docker`
- ▶ In the first boot it will also initialize it (run cloud-init, grow partition, reboot) and it will take more time

Demo: Running the instance: `limactl start ll`

Demo: Running instance: `limactl start` (MP4)

Demo: Using Docker (and more) I

- ▶ To verify that Docker is working we run the hello-world container:

```
$ docker run hello-world
```

Demo: Using Docker (and more) II

Demo: Using Docker: simple example (MP4)

Demo: Using Docker (and more) III

- ▶ Let's run something a bit more complex, nginx by sharing the mount and exposing its port to host's port 8080:

```
$ docker run -v \  
    ./static-html-directory:/usr/share/nginx/html:ro \  
    -p 8080:80 \  
    nginx
```

Demo: Using Docker (and more) IV

Demo: Using Docker: example with volume sharing and port forward (MP4)

Demo: Using Docker (and more) V

- ▶ Something even more complex... Kubernetes via kind!:

```
$ kind create cluster
```

Demo: Using Docker (and more) VI

Demo: Using Docker: running kind (MP4)

Demo: Stopping the instance: `limactl stop l`

- ▶ To stop the instance we run:
 `$ limactl stop docker`
- ▶ This will shutdown the instance.
- ▶ We can start it again via `limactl start`.

Demo: Stopping the instance: `limactl stop ll`

Demo: Stopping the instance: `limactl stop (MP4)`

Demo: Deleting the instance: `limactl delete l`

- ▶ To delete the instance we run:
`$ limactl delete docker`
- ▶ This will delete the instance so also **all its data is lost**, beware!

Demo: Deleting the instance: `limactl delete ll`

Demo: Deleting the instance: `limactl delete` (MP4)

Status, possible TODOs, help needed

- ▶ Everything packaged in pkgsrc-wip as wip/lima, wip/docker-cli and wip/kind, probably ready to be imported after pkgsrc-2025Q3
- ▶ Ubuntu, Ubuntu LTS and Debian images fails with Kernel panic - not syncing: IO-APIC + timer doesn't work! Boot with apic=debug and send a report. Then try booting with the 'noapic' option. ²
- ▶ Testing on other platforms appreciated! (in particular macOS)

²I think we have dealt with something similar for sysutils/podman!

Thanks

- ▶ Particular thank you: <maya>, <riastradh>, <ryoon>, <cirnatdan> (and I'm sure I have missed a lot of other folks, sorry!)

Conclusions

- ▶ We have introduced Lima: **Linux Machines**
- ▶ We have learned how to write/extend a custom Lima template to provision and configure packages in the guest
- ▶ We have seen how to run Docker on NetBSD via Lima with volume sharing and automatic port forwarding

References

Lima Authors.

Lima homepage.

<https://lima-vm.io/>, a.

Lima Authors.

Lima documentation.

<https://lima-vm.io/docs/>, b.

Docker Inc.

Docker Docs.

<https://docs.docker.com/>.

The Kubernetes Authors.

kind homepage.

<https://kind.sigs.k8s.io/>.

Questions?