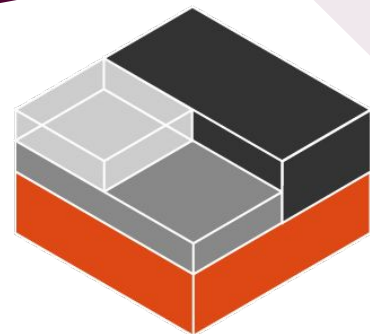


LXD: Your local cloud

Building a modern container and virtual machine manager



Stéphane Graber

LXD project leader

@stgraber

<https://stgraber.org>

stephane.graber@canonical.com

CANONICAL  ubuntu 

What are system containers?



01

They are the oldest type of containers

BSD jails, Linux vServer, Solaris Zones, OpenVZ, LXC and LXD.

02

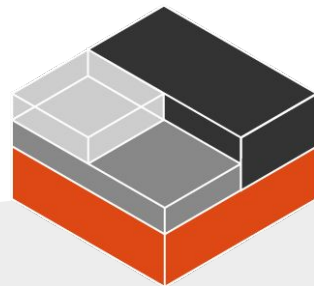
They behave like standalone systems

No need for specialized software or custom images.

03

Low overhead, easy management

Thousands can be run on one system, as easy to manage as a bunch of processes.



What are virtual machines?



01

Virtualized hardware & firmware

Behaves in many ways like a physical system.

02

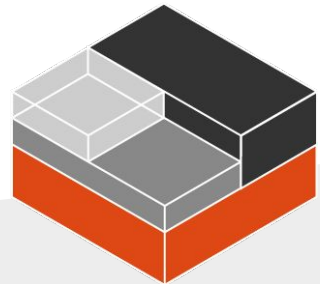
Hardware accelerated

Useful virtualization requires hardware support, additional performance gain comes from using virtualization-aware devices (e.g. virtio).

03

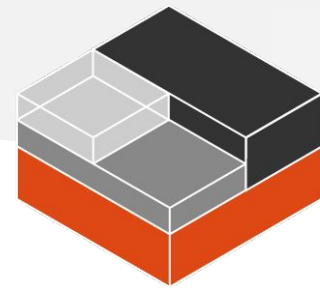
Can run just about any OS

Not constrained to Linux only.



LXD

System container
& VM manager



CLI

Ansible

Juju

OpenNebula

your own client?

LXD REST API

LXD

LXC

QEMU

Linux kernel

Host A

LXD

LXC

QEMU

Linux kernel

Host B

LXD

LXC

QEMU

Linux kernel

Host C

LXD

LXC

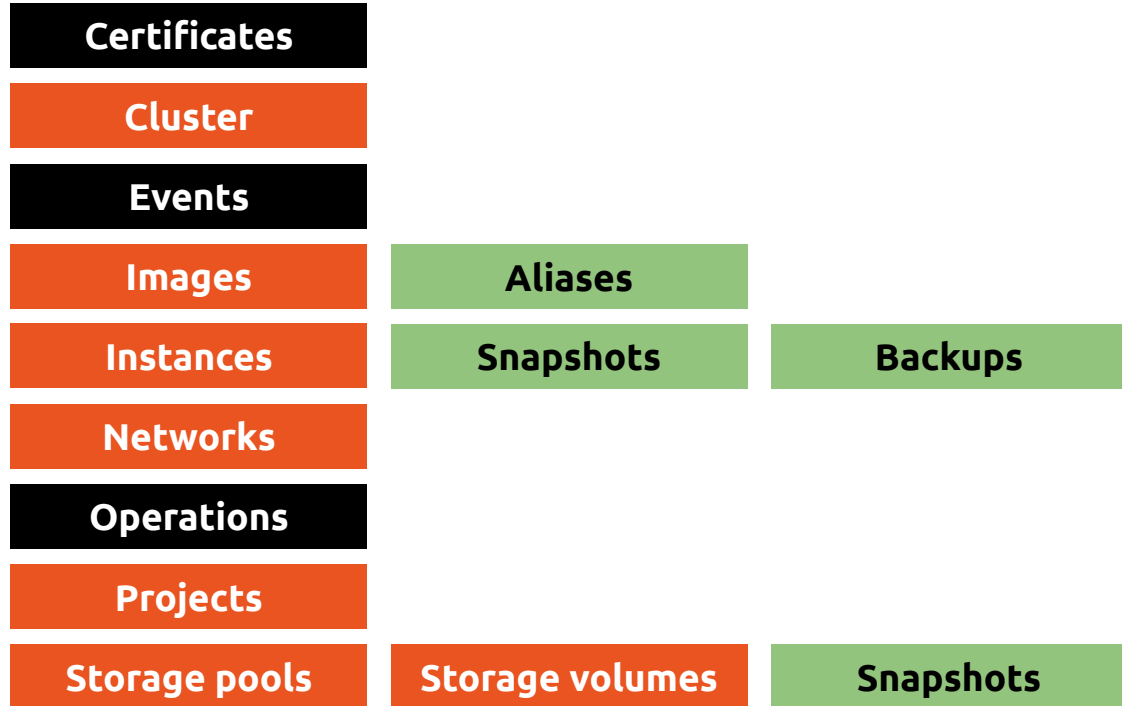
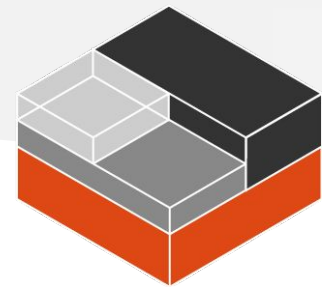
QEMU

Linux kernel

Host ...

LXD

Main components



Chromebooks



Installing Linux...

This process may take a few minutes. Starting the Linux container.



Cancel



✗ Pull Request #10276 Use large enough buffer for signature in dgst.c

#29663 failed

Chapter 21: Do not use ! to check if a pointer is NULL

Ran for 25 min

Commit 529905f

Total time 2 hrs 7 min 42 sec

#10276: Use large enough buffer for signature in dgst.c

about 3 hours ago

Branch master

nbika

[Build jobs](#)

[View config](#)

✓ # 29663.1	AMD64	</>	Compiler: gcc Xcode: xcode9.3 C	CONFIG_OPTS="" DESTDIR="_install"	5 min 30 sec
✓ # 29663.2	AMD64	</>	Compiler: gcc Xcode: xcode9.3 C	CONFIG_OPTS="no-asm -Werror --debug no-afalgeng no-shared enable-crypto-mdebug"	19 min 29 sec
✗ # 29663.3	AMD64	</>	Compiler: gcc Xcode: xcode9.3 C	CONFIG_OPTS="no-asm no-makedepend enable-buildtest-c++ --strict-warnings -D_DEP"	2 min 18 sec
✓ # 29663.4	AMD64	</>	Compiler: clang Xcode: xcode9.3 C	CONFIG_OPTS="" DESTDIR="_install"	15 min 15 sec
✓ # 29663.5	AMD64	</>	Compiler: clang Xcode: xcode9.3 C	CONFIG_OPTS="no-asm -Werror --debug no-afalgeng no-shared enable-crypto-mdebug"	24 min
✗ # 29663.6	AMD64	</>	Compiler: clang Xcode: xcode9.3 C	CONFIG_OPTS="no-asm no-makedepend enable-buildtest-c++ --strict-warnings -D_DEP"	7 min 34 sec
✓ # 29663.7	Arm64	</>	Compiler: gcc Xcode: xcode9.3 C	CONFIG_OPTS="--strict-warnings"	20 min 1 sec
✗ # 29663.8	ppc64le	</>	Compiler: clang Xcode: xcode9.3 C	CONFIG_OPTS="--strict-warnings -D_NO_STRING_INLINES"	9 min 10 sec
✓ # 29663.9	AMD64	</>	Compiler: gcc Xcode: xcode9.3 C	CONFIG_OPTS="--strict-warnings" COMMENT="Move to the BORINGTEST build when in"	4 min 34 sec
✗ # 29663.10	AMD64	</>	Compiler: clang Xcode: xcode9.3 C	CONFIG_OPTS="--strict-warnings -D_NO_STRING_INLINES no-deprecated"	1 min 34 sec
✗ # 29663.11	AMD64	</>	Compiler: clang Xcode: xcode9.3 C	CONFIG_OPTS="--strict-warnings -D_NO_STRING_INLINES no-deprecated" BUILDONL	1 min 50 sec
✓ # 29663.12	AMD64	</>	Compiler: i686-w64-mingw32-gcc Xcode: xcode9.3 C	CONFIG_OPTS="no-stdio" BUILDONLY="yes"	6 min 29 sec

LXD system containers



01

Comprehensive and flexible resource limits

CPU, memory, network, storage, processes and various kernel limits

02

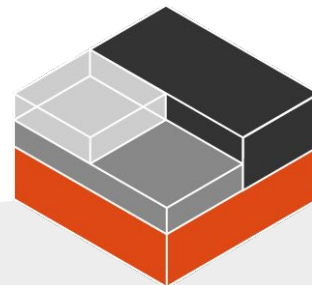
Device passthrough

Network, GPU, USB, disks and arbitrary unix char/block devices

03

Advanced features

System call interception, uevent injection, isolated id maps, shiftfs, ...



LXD virtual machines



01

Modern machines

UEFI with Secure Boot (where available), virtio devices only, based on QEMU 4.2+.

02

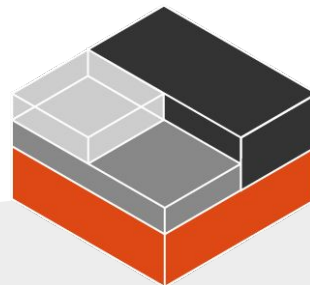
Same API and semantics as our containers

No particular VM knowledge needed by existing clients.

03

Integrates seamlessly with LXD networks, storage, projects, profiles, ...

All existing configuration can be shared between containers and virtual machines, profiles with resource limits or devices can apply to both types.



LXD clustering



01

Built-in clustering support

No external dependencies, all LXD 3.0 or higher installations can be instantly turned into a cluster.

02

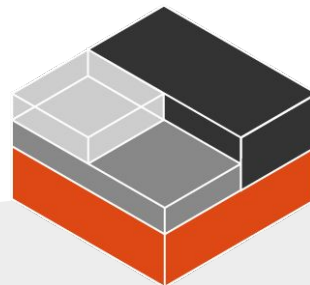
Same API as a single node

Clients that aren't clustering aware just see it as a very large LXD instance.

03

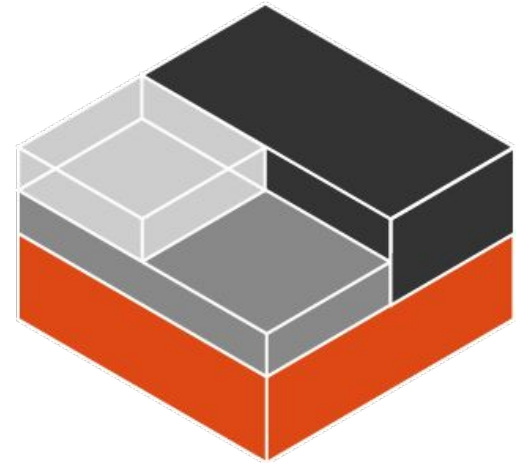
Scales to thousands of containers on dozens of nodes

Uses a built-in distributed database and cross-connections between the nodes to offer a consistent view to clients and load-balance containers.





Demo time!



Let's recap



01

System containers or virtual machines

*Pick what's best suited for the workload, management is identical.
Wide variety of images available for either type.*

02

Seamless scaling with clustering

*Single entity to manage, highly available and easily scalable.
Combined with CEPH, allows for fault tolerance.*

03

Easy storage, network, GPU and generic device passthrough

*Expose hardware directly to your containers.
Use quotas and limits to prevent abuse.*

04

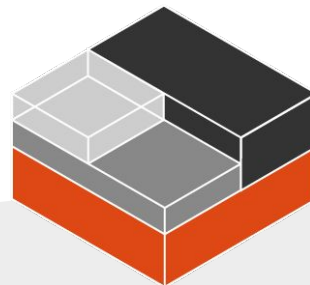
Works everywhere

*Runs on a wide variety of Linux distributions and on all mainstream architectures.
Client also available for Windows and macOS.*

05

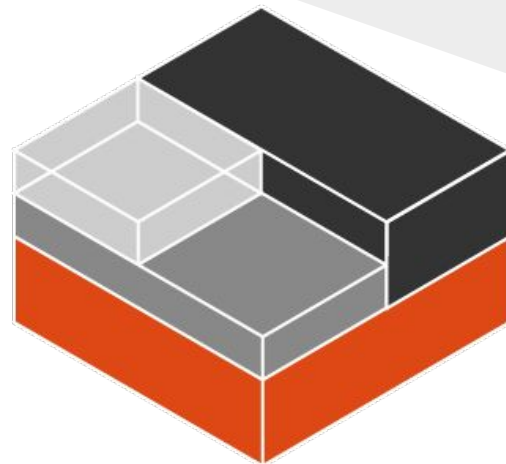
Production ready

*Long term support releases with 5 years of support.
LXD has been around for over 4 years, LXC for over a decade.*



Questions ?

Website: <https://linuxcontainers.org/lxd>
Code: <https://github.com/lxc/lxd>
Online demo: <https://linuxcontainers.org/lxd/try-it>
Community: <https://discuss.linuxcontainers.org>



Stéphane Graber

LXD project leader

@stgraber

<https://stgraber.org>

stephane.graber@canonical.com

