System containers at scale

An introduction to LXD clustering

Stéphane Graber
LXD project leader
@stgraber
https://stgraber.org
stephane.graber@canonical.com
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.*
LXD
System container manager

LXD
CLI
Ansible
Juju
OpenNebula
your own client?

LXD REST API

LXD
LXC
Linux kernel
Host A

LXD
LXC
Linux kernel
Host B

LXD
LXC
Linux kernel
Host C

LXD
LXC
Linux kernel
Host ...

LXD
Linux kernel

Host A

Host B

Host C

Host ...
What LXD is

01
Simple
Clean command line interface, simple REST API and clear terminology.

02
Fast
Image based, optimized storage & migration, direct hardware access.

03
Secure
Safe by default. Combines all available kernel security features.

04
Scalable
From a single container on a laptop to tens of thousands of containers in a cluster.
Wide selection of images

Updated daily
Installing Linux...

This process may take a few minutes. Starting the Linux container.
# Travis-CI

## openssl / openssl

### Build jobs

<table>
<thead>
<tr>
<th>Job</th>
<th>Status</th>
<th>Build Type</th>
<th>Build Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>#29663.1</td>
<td>✓</td>
<td>AM064</td>
<td>5 min 30 sec</td>
</tr>
<tr>
<td>#29663.2</td>
<td>✓</td>
<td>AM064</td>
<td>19 min 29 sec</td>
</tr>
<tr>
<td>#29663.3</td>
<td>✗</td>
<td>AM064</td>
<td>2 min 18 sec</td>
</tr>
<tr>
<td>#29663.4</td>
<td>✓</td>
<td>AM064</td>
<td>15 min 15 sec</td>
</tr>
<tr>
<td>#29663.5</td>
<td>✓</td>
<td>AM064</td>
<td>24 min</td>
</tr>
<tr>
<td>#29663.6</td>
<td>✗</td>
<td>AM064</td>
<td>7 min 34 sec</td>
</tr>
<tr>
<td>#29663.7</td>
<td>✓</td>
<td>AMv64</td>
<td>20 min 1 sec</td>
</tr>
<tr>
<td>#29663.8</td>
<td>✗</td>
<td>AMv64</td>
<td>9 min 10 sec</td>
</tr>
<tr>
<td>#29663.9</td>
<td>✓</td>
<td>Compiler: gcc Xcode: xcode9.3.C</td>
<td>4 min 34 sec</td>
</tr>
<tr>
<td>#29663.10</td>
<td>✓</td>
<td>Compiler: clang Xcode: xcode9.3.C</td>
<td>1 min 34 sec</td>
</tr>
<tr>
<td>#29663.11</td>
<td>✗</td>
<td>Compiler: clang Xcode: xcode9.3.C</td>
<td>1 min 50 sec</td>
</tr>
<tr>
<td>#29663.12</td>
<td>✓</td>
<td>Compiler: i686-w64-mingw32-gcc Xcode: xcode9.3.C</td>
<td>6 min 29 sec</td>
</tr>
</tbody>
</table>

### View config

<table>
<thead>
<tr>
<th>Config Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONFIG_OPTS=&quot;&quot;</td>
<td>DESTDIR=&quot;.install&quot;</td>
</tr>
<tr>
<td>CONFIG_OPTS=&quot;&quot;</td>
<td>&quot;no-asm -Werror --debug -no-afalgeng no-shared enable-crypto-mdebug&quot;</td>
</tr>
<tr>
<td>CONFIG_OPTS=&quot;&quot;</td>
<td>&quot;no-asm no-makedepend enable-buildest -c++ -strict-warnings no&gt;Delete&quot;</td>
</tr>
<tr>
<td>CONFIG_OPTS=&quot;&quot;</td>
<td>&quot;-strict-warnings&quot;</td>
</tr>
<tr>
<td>CONFIG_OPTS=&quot;&quot;</td>
<td>&quot;-strict-warnings -D _NO_STRING_INLINES&quot;</td>
</tr>
<tr>
<td>CONFIG_OPTS=&quot;&quot;</td>
<td>&quot;strict warnings&quot; COMMENT=&quot;&quot;Move to the BORINGTEST build when in&quot;</td>
</tr>
<tr>
<td>CONFIG_OPTS=&quot;&quot;</td>
<td>&quot;strict warnings -D _NO_STRING_INLINES re-deprecated&quot;</td>
</tr>
<tr>
<td>CONFIG_OPTS=&quot;&quot;</td>
<td>&quot;strict warnings -D _NO_STRING_INLINES no-deprecated&quot;</td>
</tr>
<tr>
<td>CONFIG_OPTS=&quot;&quot;</td>
<td>&quot;strict warnings&quot;</td>
</tr>
<tr>
<td>CONFIG_OPTS=&quot;&quot;</td>
<td>&quot;no-stdio BUILDONLY=&quot;yes&quot;&quot;</td>
</tr>
</tbody>
</table>

### Pull Request #30276

Use large enough buffer for signature in dgst.c

Chapter 25: Do not use ! to check if a pointer is NULL

- Commit 5299865f
- #10276: Use large enough buffer for signature in dgst.c
- Branch master

nbika

### #29663 failed

- Ran for 25 min
- Total time 2 hrs 7 min 42 sec
- about 3 hours ago
LXD Main components

Certificates
Cluster
Events
Images
Instances
Networks
Operations
Projects
Storage pools

Aliases
Snapshots
Backups
Storage volumes
Snapshots
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!
LXD everywhere

**Linux**
- `snap install lxd`
  - requires snapd on a supported Linux distribution
- **Native packages**
  - available for some releases on Alpine, ArchLinux, Fedora, Gentoo, OpenSUSE and Ubuntu
- **On your Chromebook**
  - Search for “Terminal” in your app launcher

**MacOS**
- `brew install lxc`
  - requires Homebrew on current MacOS

**Windows**
- `choco install lxc`
  - requires Chocolatey on current Windows 10
Contributing to LXD

01 **Written in Go**
With low level logic in C through a variety of libraries.

02 **Fully translatable client**
An easy way to contribute to LXD, translate our CLI in your language!

03 **API client libraries**
Official ones for Go and Python
Additional ones in Ruby, Node, Java, Haskell, ...

04 **Apache2 licensed**
Re-use and improve any of our Go packages in your own projects.

05 **No copyright assignment**
Easy contributions, no legal paperwork, just send a pull request.

06 **Online user community**
Very active discussion forum with active experts in container networking, security and more.

git clone https://github.com/lxc/lxd
We have stickers, come get them in front!

Questions?

Stéphane Graber
LXD project leader

@stgraber
https://stgraber.org
stephane.graber@canonical.com

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