

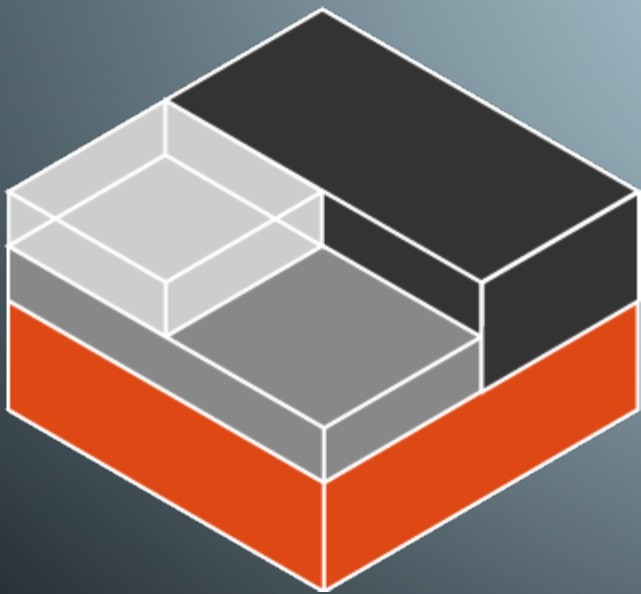
LXC VS VAGRANT

POR:

RUBÉN CALVO VILLAZÁN

JOSE LUIS ROJANO RAMÍREZ

¿QUÉ ES LXD?



Es un gestor de contenedores de sistemas de próxima generación. Ofrece una experiencia de usuario similar a las máquinas virtuales, pero usando contenedores de Linux en su lugar. Basado en imágenes, está construido alrededor de una API REST muy poderosa pero bastante simple. Este proyecto fue fundado y actualmente dirigido por Canonical LTD con contribuciones de otras compañías y contribuyentes individuales.

¿CÓMO INSTALAR LXD?

LXD no es una reescritura de LXC, de hecho, se construye sobre LXC para brindar una nueva y mejor experiencia de usuario. Bajo el capó, LXD usa LXC a través de liblxc y su enlace Go para crear y administrar los contenedores.

Los dispositivos están virtualizados de forma que cada contenedor ve sus propios dispositivos. Los procesos que se ejecutan son nativos del equipo host y no es necesario crear un hardware virtualizado completo, por lo que el funcionamiento global es más eficiente.

En este caso lo haremos con Ubuntu Server, ya tiene instalado por defecto LXD:

Creamos un contenedor con Ubuntu:

```
jrr@jrr:~$ lxc launch ubuntu: c1
Generating a client certificate. This may take a minute...
If this is your first time using LXD, you should also run: sudo lxd init
To start your first container, try: lxc launch ubuntu:16.04

Creando c1
Iniciando c1
jrr@jrr:~$ _
```

Comprobamos que está funcionando:

```
jrr@jrr:~$ lxc list
+-----+-----+-----+-----+-----+-----+
| NOMBRE | ESTADO | IPV4 | IPV6 | TIPO | SNAPSHOTS |
+-----+-----+-----+-----+-----+-----+
| c1     | RUNNING |      |      | PERSISTENT | 0          |
+-----+-----+-----+-----+-----+-----+
```

Accedemos al contenedor c1 y comprobamos la versión de

```
jrr@jrr:~$ lxc exec c1 bash
root@c1:~# lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 18.04.1 LTS
Release:        18.04
Codename:       bionic
root@c1:~#
```


Una vez dentro ya podemos hacer lo que quedamos con él.
Vamos a crear otro contendor con Centos en este caso:

```
root@jrr:/# lxc launch images:centos/7 c2
Generating a client certificate. This may take a minute...
If this is your first time using LXD, you should also run: sudo lxd init
To start your first container, try: lxc launch ubuntu:16.04

Creando c2
Iniciando c2
root@jrr:/#
```

Listamos para comprobar que están los dos instalados y

```
+-----+-----+-----+-----+-----+
| NOMBRE | ESTADO | IPV4 | IPV6 | TIPO | SNAPSHOTS |
+-----+-----+-----+-----+-----+
| c1     | RUNNING |      |      | PERSISTENT | 0          |
+-----+-----+-----+-----+-----+
| c2     | RUNNING |      |      | PERSISTENT | 0          |
+-----+-----+-----+-----+-----+
```

```
root@jrr:/# lxc stop c2
root@jrr:/# lxc delete c2
root@jrr:/# lxc list
```

```
+-----+-----+-----+-----+-----+
| NOMBRE | ESTADO | IPV4 | IPV6 | TIPO | SNAPSHOTS |
+-----+-----+-----+-----+-----+
| c1     | RUNNING |      |      | PERSISTENT | 0          |
+-----+-----+-----+-----+-----+
root@jrr:/# _
```

contenedores:

Vamos a configurar la red :

Primero en el Host con ifconfig localizamos la red de lxd:

```
enp0s8  Link encap:Ethernet direcciónHW 08:00:27:ae:72:9b
        Direc. inet:192.168.56.105 Difus.:192.168.56.255 Másc:255.255.255.0
        Dirección inet6: fe80::a00:27ff:feae:729b/64 Alcance:Enlace
        ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
        Paquetes RX:9 errores:0 perdidos:0 overruns:0 frame:0
        Paquetes TX:956 errores:0 perdidos:0 overruns:0 carrier:0
        colisiones:0 long.colatX:1000
        Bytes RX:828 (828.0 B) TX bytes:57528 (57.5 KB)

lo      Link encap:Bucle local
        Direc. inet:127.0.0.1 Másc:255.0.0.0
        Dirección inet6: ::1/128 Alcance:Anfitrión
        ACTIVO BUCLE FUNCIONANDO MTU:65536 Métrica:1
        Paquetes RX:4961 errores:0 perdidos:0 overruns:0 frame:0
        Paquetes TX:4961 errores:0 perdidos:0 overruns:0 carrier:0
        colisiones:0 long.colatX:1
        Bytes RX:400188 (400.1 KB) TX bytes:400188 (400.1 KB)

lxdbr0  Link encap:Ethernet direcciónHW fe:b0:05:3f:20:05
        Direc. inet:192.168.57.100 Difus.:192.168.57.255 Másc:255.255.255.0
        Dirección inet6: fe80::c8d6:b8ff:feb7:da88/64 Alcance:Enlace
        Dirección inet6: fe80::1/64 Alcance:Enlace
        ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
        Paquetes RX:29 errores:0 perdidos:0 overruns:0 frame:0
        Paquetes TX:16 errores:0 perdidos:0 overruns:0 carrier:0
        colisiones:0 long.colatX:1000
        Bytes RX:1516 (1.5 KB) TX bytes:1460 (1.4 KB)

vethIH6BU6 Link encap:Ethernet direcciónHW fe:b0:05:3f:20:05
        Dirección inet6: fe80::fcb0:5ff:fe3f:2005/64 Alcance:Enlace
        ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
        Paquetes RX:29 errores:0 perdidos:0 overruns:0 frame:0
        Paquetes TX:19 errores:0 perdidos:0 overruns:0 carrier:0
        colisiones:0 long.colatX:1000
        Bytes RX:1922 (1.9 KB) TX bytes:1638 (1.6 KB)

jrr@jrr:~$
```

A continuación vamos a darle una ip nueva 192.168.57.100

Hacemos lo mismo en c1 pero a eth0 y con la ip

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.57.105 netmask 255.255.255.0 broadcast 192.168.57.255
    inet6 fe80::216:3eff:fec0:a6dd prefixlen 64 scopeid 0x20<link>
    ether 00:16:3e:c0:a6:dd txqueuelen 1000 (Ethernet)
    RX packets 19 bytes 1638 (1.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 29 bytes 1922 (1.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1 (Local Loopback)
    RX packets 29 bytes 2412 (2.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 29 bytes 2412 (2.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@c1:~# _
```

Y ya estarían conectadas entre sí:

```
root@c1:~# ping 192.168.57.100
PING 192.168.57.100 (192.168.57.100) 56(84) bytes of data.
64 bytes from 192.168.57.100: icmp_seq=1 ttl=64 time=0.114 ms
64 bytes from 192.168.57.100: icmp_seq=2 ttl=64 time=0.183 ms
64 bytes from 192.168.57.100: icmp_seq=3 ttl=64 time=0.131 ms
64 bytes from 192.168.57.100: icmp_seq=4 ttl=64 time=0.135 ms
64 bytes from 192.168.57.100: icmp_seq=5 ttl=64 time=0.156 ms
64 bytes from 192.168.57.100: icmp_seq=6 ttl=64 time=0.132 ms
64 bytes from 192.168.57.100: icmp_seq=7 ttl=64 time=0.126 ms
64 bytes from 192.168.57.100: icmp_seq=8 ttl=64 time=0.168 ms
64 bytes from 192.168.57.100: icmp_seq=9 ttl=64 time=0.157 ms
64 bytes from 192.168.57.100: icmp_seq=10 ttl=64 time=0.146 ms
64 bytes from 192.168.57.100: icmp_seq=11 ttl=64 time=0.132 ms
64 bytes from 192.168.57.100: icmp_seq=12 ttl=64 time=0.135 ms
64 bytes from 192.168.57.100: icmp_seq=13 ttl=64 time=0.134 ms
64 bytes from 192.168.57.100: icmp_seq=14 ttl=64 time=0.118 ms
64 bytes from 192.168.57.100: icmp_seq=15 ttl=64 time=0.134 ms
64 bytes from 192.168.57.100: icmp_seq=16 ttl=64 time=0.132 ms
64 bytes from 192.168.57.100: icmp_seq=17 ttl=64 time=0.136 ms
64 bytes from 192.168.57.100: icmp_seq=18 ttl=64 time=0.142 ms
64 bytes from 192.168.57.100: icmp_seq=19 ttl=64 time=0.135 ms
^C
--- 192.168.57.100 ping statistics ---
19 packets transmitted, 19 received, 0% packet loss, time 18001ms
rtt min/avg/max/mdev = 0.114/0.139/0.183/0.018 ms
```


¿QUÉ ES VAGRANT?



Vagrant es una herramienta gratuita de línea de comandos que permite generar entornos de desarrollo reproducibles y compartibles de forma muy sencilla. Para ello, Vagrant crea y configura máquinas virtuales a partir de simples ficheros de configuración.

Basta con compartir el fichero de configuración de Vagrant (“Vagrantfile”) lo que asegura que todos los desarrolladores tienen el mismo entorno, con las mismas dependencias y configuración. Con Vagrant, compartir pesadas máquinas virtuales o el ya mítico “en mi ordenador funciona” son cosas del pasado.

Además, dado que la configuración de la máquina virtual es un simple fichero de texto plano, podemos incluir este fichero en nuestro repositorio en el control de versiones, junto con el resto del código del proyecto. De esta manera, un nuevo desarrollador que se incorpore al equipo simplemente tendrá que clonar el repositorio del proyecto y ejecutar Vagrant para tener el entorno de desarrollo montado y funcionando en cuestión de minutos.

¿CÓMO INSTALAR VAGRANT?

En Ubuntu Server instalamos virtual box y vagrant con sudo apt-get install virtualbox\Vagrant e iniciamos con vagrant init,

```
root@valkyrie:ISE# ls
Vagrantfile
```

```
root@valkyrie:ISE# cat Vagrantfile
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/trusty64"
end
root@valkyrie:ISE#
```

Instalación de Ubuntu sería así:

Ejecutar Vagrant up para crear la máquina a partir del

```
root@valkyrie:ISE# vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'ubuntu/trusty64'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'ubuntu/trusty64' is up to date...
==> default: Setting the name of the VM: ISE_default_1545045285693_4219
==> default: Clearing any previously set forwarded ports...
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2222
```


Nos conectamos a la máquina virtual con vagrant ssh:

```
root@valkyrie:ISE# vagrant ssh
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 3.13.0-163-generic x86_64)

* Documentation:  https://help.ubuntu.com/

System information as of Mon Dec 17 11:15:13 UTC 2018

System load:  0.83               Processes:            83
Usage of /:   3.6% of 39.34GB    Users logged in:     0
Memory usage: 25%               IP address for eth0: 10.0.2.15
Swap usage:   0%

Graph this data and manage this system at:
https://landscape.canonical.com/

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

New release '16.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

vagrant@vagrant-ubuntu-trusty-64:~$
```

```
vagrant@vagrant-ubuntu-trusty-64:~$ pwd
/home/vagrant
vagrant@vagrant-ubuntu-trusty-64:~$ whoami
vagrant
vagrant@vagrant-ubuntu-trusty-64:~$
```

Una vez que hayamos terminado de trabajar con la máquina podemos ejecutar los siguientes comandos:

‘**vagrant suspend**’: Pausa la máquina virtual, guardando el estado actual en el disco duro. Permite arrancar de nuevo la máquina muy rápidamente con `vagrant up` con el estado exacto en el que se quedó.

‘**vagrant halt**’: Realiza un apagado controlado de la máquina virtual (igual a apagar una máquina física). Como en el caso anterior, podemos volver a arrancar la máquina virtual con `vagrant up`, aunque en este caso el arranque es más lento que al hacer un “suspend” (ya que tiene que volver a iniciar el sistema operativo).

‘**vagrant destroy**’: Destruye la máquina virtual y todo su contenido.

Probamos la conexión:

```
[vagrant@centos1 ~]$ ping 192.168.12.1
PING 192.168.12.1 (192.168.12.1) 56(84) bytes of data.
64 bytes from 192.168.12.1: icmp_seq=1 ttl=64 time=0.294 ms
64 bytes from 192.168.12.1: icmp_seq=2 ttl=64 time=0.408 ms
^C
--- 192.168.12.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.294/0.351/0.408/0.057 ms
```

```
root@valkyrie:ISE# ping 192.168.12.11
PING 192.168.12.11 (192.168.12.11) 56(84) bytes of data.
64 bytes from 192.168.12.11: icmp_seq=1 ttl=64 time=0.533 ms
64 bytes from 192.168.12.11: icmp_seq=2 ttl=64 time=0.489 ms
^C
--- 192.168.12.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 14ms
rtt min/avg/max/mdev = 0.489/0.511/0.533/0.022 ms
```


También podemos configurar a la vez varias máquinas en el mismo `vagrantfile`:

```
Vagrant.configure("2") do |config|
  config.vm.provision "shell", inline: "echo Hello"

  config.vm.define "web" do |web|
    web.vm.box = "apache"
  end

  config.vm.define "db" do |db|
    db.vm.box = "mysql"
  end
end
```


COMPARATIVA LXD VS VAGRANT

- LXD:

- Más simple
- Código Abierto
- Dispone de API
- Más rápido
- Aislamiento completo del SO

- Vagrant

- Requiere configurar vagrantfile
- Muy pesado
- Facilidad a la hora de compartir trabajo
- Hace falta instalar virtual box
- No hay aislamiento del SO (carpeta vinculada /vagrant)

COMPAÑÍAS QUE LOS UTILIZAN

- LXD:



Stockopedia



Ocado Technology



tutuf



Dek-D

- Vagrant:



HashiCorp



Zscaler



Intellectaco









Skydive project

POPULARIDAD

LXD:

Vagrant:

	<div>Fans 58</div> <div>Jobs 3</div> <div>Votes 21</div>	<div>Fans 24</div> <div>Jobs 1</div> <div>Votes 2</div>
Hacker News, Reddit, Stack Overflow Stats	<div><div>Y</div><div>-</div></div> <div><div></div><div>223</div></div> <div><div></div><div>97</div></div>	<div><div>Y</div><div>35</div></div> <div><div></div><div>51</div></div> <div><div></div><div>0</div></div>
GitHub Stats	<div><div></div><div><div>★</div>2.05K</div><div><div>🔗</div>479</div><div><div></div>about 11 hours ago</div></div>	<div>No public GitHub repository stats available</div>

REFERENCIAS

- <https://linuxcontainers.org/lxd/introduction/>
- <https://elpuig.xeill.net/Members/vcarceler/articulos/contenedores-con-lxd-lxc>
- <https://stackshare.io/stackups/lxd-vs-vagrant-cloud>
- https://www.reddit.com/r/devops/comments/5yngl6/lxdock_be_cause_vagrant_is_too_heavy/
- <https://www.vagrantup.com/intro/index.html>
- <https://www.vagrantup.com/docs/multi-machine/>