

# Menginstall LXD di Ubuntu 20.04

Nah, karena akan ada kebutuhan menggunakan lxc, pada catatan kali ini saya akan memasang lxd untuk menggunakan lxc. Untuk mengetahui perbedaan antara lxc & lxd bisa baca referensi di bawah [1]. Oke, langsung saja berikut langkah-langkahnya.

Asumsikan kita telah memiliki server/VPS dengan sistem operasi Ubuntu 20.04 (fresh install, alias VPS baru dibuat), dengan begitu kita tinggal remote server/VPS tersebut dengan ssh:

```
ssh root@10.x.x.x -p2020
```

Setelah berhasil masuk, yang pertama perlu dilakukan adalah memperbarui cache repo dengan apt update. Selanjutnya lakukan instalasi lxd dengan perintah berikut:

```
apt install lxd
```

CATATAN: di sini saya tidak menggunakan sudo karena saya sudah menggunakan akun root. Jika Anda menggunakan user biasa, silakan tambahkan sudo di depan perintah tadi.

Contoh outputnya seperti berikut :

```
root@nacita-riset:~# apt install lxd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  liblzo2-2 snapd squashfs-tools
Suggested packages:
  zenity | kdialog
The following NEW packages will be installed:
  liblzo2-2 lxd snapd squashfs-tools
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 30.7 MB of archives.
After this operation, 134 MB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://asi-fs-n.contabo.net/ubuntu focal/main amd64 liblzo2-2 amd64
2.10-2 [50.8 kB]
Get:2 http://asi-fs-n.contabo.net/ubuntu focal/main amd64 squashfs-tools
amd64 1:4.4-1 [121 kB]
Get:3 http://asi-fs-n.contabo.net/ubuntu focal-updates/main amd64 snapd
amd64 2.49.2+20.04 [30.6 MB]
Get:4 http://asi-fs-n.contabo.net/ubuntu focal/universe amd64 lxd all 1:0.9
[5444 B]
Fetched 30.7 MB in 1s (24.4 MB/s)
Preconfiguring packages ...
Selecting previously unselected package liblzo2-2:amd64.
(Reading database ... 63931 files and directories currently installed.)
Preparing to unpack .../liblzo2-2_2.10-2_amd64.deb ...
Unpacking liblzo2-2:amd64 (2.10-2) ...
```

```
Selecting previously unselected package squashfs-tools.
Preparing to unpack .../squashfs-tools_1%3a4.4-1_amd64.deb ...
Unpacking squashfs-tools (1:4.4-1) ...
Selecting previously unselected package snapd.
Preparing to unpack .../snapd_2.49.2+20.04_amd64.deb ...
Unpacking snapd (2.49.2+20.04) ...
Setting up liblzo2-2:amd64 (2.10-2) ...
Setting up squashfs-tools (1:4.4-1) ...
Setting up snapd (2.49.2+20.04) ...
Created symlink /etc/systemd/system/multi-
user.target.wants/snapd.apparmor.service →
/lib/systemd/system/snapd.apparmor.service.
Created symlink /etc/systemd/system/multi-
user.target.wants/snapd.autoimport.service →
/lib/systemd/system/snapd.autoimport.service.
Created symlink /etc/systemd/system/multi-
user.target.wants/snapd.core-
fixup.service → /lib/systemd/system/snapd.core-
fixup.service.
Created symlink /etc/systemd/system/multi-
user.target.wants/snapd.recovery-
chooser-trigger.service → /lib/systemd/system/snapd.recovery-
chooser-
trigger.service.
Created symlink /etc/systemd/system/multi-
user.target.wants/snapd.seeded.service →
/lib/systemd/system/snapd.seeded.service.
Created symlink /etc/systemd/system/cloud-
final.service.wants/snapd.seeded.service →
/lib/systemd/system/snapd.seeded.service.
Created symlink /etc/systemd/system/multi-
user.target.wants/snapd.service →
/lib/systemd/system/snapd.service.
Created symlink /etc/systemd/system/timers.target.wants/snapd.snap-
repair.timer → /lib/systemd/system/snapd.snap-
repair.timer.
Created symlink /etc/systemd/system/sockets.target.wants/snapd.socket →
/lib/systemd/system/snapd.socket.
Created symlink /etc/systemd/system/final.target.wants/snapd.system-
shutdown.service → /lib/systemd/system/snapd.system-
shutdown.service.
snapd.failure.service is a disabled or a static unit, not starting it.
snapd.snap-repair.service is a disabled or a static unit, not starting it.
Selecting previously unselected package lxd.
(Reading database ... 64042 files and directories currently installed.)
Preparing to unpack .../archives/lxd_1%3a0.9_all.deb ...
=> Installing the LXD snap
==> Checking connectivity with the snap store
==> Installing the LXD snap from the 4.0 track for ubuntu-20.04
2021-06-13T05:34:44+02:00 INFO Waiting for automatic snapd restart...
Warning: /snap/bin was not found in your $PATH. If you've not restarted your
session since you
    installed snapd, try doing that. Please see
https://forum.snapcraft.io/t/9469 for more
    details.

lxd (4.0/stable) 4.0.6 from Canonical✓ installed
=> Snap installation complete
```

```
==> Cleaning up leftovers
Failed to stop lxd.socket: Unit lxd.socket not loaded.
Failed to stop lxd.service: Unit lxd.service not loaded.
Failed to stop lxd-containers.service: Unit lxd-containers.service not
loaded.
Failed to disable unit: Unit file lxd.socket does not exist.
Unpacking lxd (1:0.9) ...
Setting up lxd (1:0.9) ...
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
Processing triggers for man-db (2.9.1-1) ...
```

CATATAN: Jika diperhatikan pada output di atas, bahwa pada Ubuntu 20.04 kita dipaksa untuk menggunakan snap. Baiklah, walau demikian kita masih dapat menggunakannya. Mungkin jika Anda **TIDAK SUKA** menggunakan snap, mungkin bisa coba gunakan cara lain. Namun pada catatan saya ini tidak akan membahasnya.

Lanjut ke langkah berikutnya, karena saya lihat paket `bridge-utils` belum dibawa oleh paket `lxd` tadi, maka akan saya pasang terlebih dahulu:

```
apt install bridge-utils
```

Apa fungsinya, barangkali Anda bertanya. Paket tersebut digunakan oleh sistem Linux untuk membuat bridge interface yang akan digunakan oleh `lxc`/container nantinya.

Setelah proses instalasi paket tersebut di atas selesai, kita lanjut ke inisiasi LXD dengan perintah `lxd init`, contoh keluarannya adalah sebagai berikut:

CATATAN: tekan tombol ENTER untuk menerima nilai default, atau ketik pilihan yang ditawarkan baru tekan ENTER

```
root@nacita-riset:~# lxd init
Would you like to use LXD clustering? (yes/no) [default=no]:
Do you want to configure a new storage pool? (yes/no) [default=yes]:
Name of the new storage pool [default=default]:
Name of the storage backend to use (ceph, btrfs, dir, lvm, zfs)
[default=zfs]:
Create a new ZFS pool? (yes/no) [default=yes]:
Would you like to use an existing empty block device (e.g. a disk or
partition)? (yes/no) [default=no]:
Size in GB of the new loop device (1GB minimum) [default=30GB]:
Would you like to connect to a MAAS server? (yes/no) [default=no]:
Would you like to create a new local network bridge? (yes/no) [default=yes]:
What should the new bridge be called? [default=lxdbr0]:
What IPv4 address should be used? (CIDR subnet notation, "auto" or "none")
[default=auto]:
What IPv6 address should be used? (CIDR subnet notation, "auto" or "none")
[default=auto]: none
Would you like the LXD server to be available over the network? (yes/no)
[default=no]:
```

```
Would you like stale cached images to be updated automatically? (yes/no)
[default=yes] no
Would you like a YAML "lxd init" preseed to be printed? (yes/no)
[default=no]:
```

Dengan demikian, lxd kita siap digunakan untuk menjalankan linux container (lxc) VM kita. Namun, kita cek dulu, mestinya akan ada interface baru berupa lxdbr0 seperti yang kita minta saat menjalankan lxd init tadi. Kita coba konfirmasi dengan perintah ip a:

```
oot@nacita-riset:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 00:50:56:42:a3:62 brd ff:ff:ff:ff:ff:ff
    inet 95.xxx.xxx.xxx/32 scope global eth0
<----- yang di sini ip publik, saya umpetin aja yak...
        valid_lft forever preferred_lft forever
3: lxdbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state
DOWN group default qlen 1000
    link/ether 00:16:3e:2d:4b:db brd ff:ff:ff:ff:ff:ff
    inet 10.104.166.1/24 scope global lxdbr0
<----- nah, ini interface baru yang dibuat
        valid_lft forever preferred_lft forever
```

Coba kita buat container pertama kita dengan perintah lxc:

```
lxc launch ubuntu:20.04 focal
```

Tunggu beberapa saat, container image ubuntu:20.04 selesai diunduh dan lxc akan membuat container baru dengan nama focal. Tentu variabel focal di sini dapat diganti apa saja sesuai kebutuhan, karena itu adalah nama containernya.

```
root@nacita-riset:~# lxc list
+-----+-----+-----+-----+-----+-----+
| NAME   | STATE |          IPV4          |   IPV6   | TYPE   | SNAPSHOTS |
+-----+-----+-----+-----+-----+-----+
| focal  | RUNNING | 10.104.166.79 (eth0) |          | CONTAINER | 0          |
+-----+-----+-----+-----+-----+-----+
```

Perintah di atas untuk menampilkan daftar container yang telah dibuat. Di sini saya memiliki satu buah container dengan nama focal. Kita coba masuk ke dalam containernya dengan perintah lxc exec

```
root@nacita-riset:~# lxc exec focal bash
root@focal:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
```

```

default qlen 1000
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
  inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
  inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
4: eth0@if5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state
UP group default qlen 1000
  link/ether 00:16:3e:06:e2:e7 brd ff:ff:ff:ff:ff:ff link-netnsid 0
  inet 10.104.166.79/24 brd 10.104.166.255 scope global dynamic eth0
    valid_lft 3589sec preferred_lft 3589sec
  inet6 fe80::216:3eff:fe06:e2e7/64 scope link
    valid_lft forever preferred_lft forever
root@focal:~#

```

Bisa dilihat bahwa saya sudah berhasil masuk ke dalam container, lihat pada perbedaan hostname & alamat IP yang ada di dalam container. Sampai di sini catatannya berakhir, namun saya akan menambahkan sedikit keluaran perintah lanjutan untuk membuat container baru dengan sistem operasi CentOS 7 seperti berikut:

```

root@focal:~# exit
root@nacita-riset:~# lxc launch images:centos/7 centos7
Creating centos7
Starting centos7
root@nacita-riset:~# lxc list
+-----+-----+-----+-----+-----+-----+
| NAME   | STATE |          IPV4          | IPV6 |  TYPE  | SNAPSHOTS |
+-----+-----+-----+-----+-----+-----+
| centos7 | RUNNING | 10.104.166.129 (eth0) |      | CONTAINER | 0          |
+-----+-----+-----+-----+-----+-----+
| focal   | RUNNING | 10.104.166.79 (eth0)  |      | CONTAINER | 0          |
+-----+-----+-----+-----+-----+-----+
root@nacita-riset:~# lxc exec centos7 bash
[root@centos7 ~]# ls
[root@centos7 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
  inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
  inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
6: eth0@if7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state
UP group default qlen 1000
  link/ether 00:16:3e:16:a3:51 brd ff:ff:ff:ff:ff:ff link-netnsid 0
  inet 10.104.166.129/24 brd 10.104.166.255 scope global dynamic eth0
    valid_lft 3588sec preferred_lft 3588sec
  inet6 fe80::216:3eff:fe16:a351/64 scope link
    valid_lft forever preferred_lft forever
[root@centos7 ~]# exit

```

```
root@nacita-riset:~#
```

TAMBAHAN: membuat container dengan spesifikasi tertentu memanfaatkan instance type aws

```
lxc launch ubuntu:20.04 my-instance -t t2.micro
```

Demikian catatan kali ini, semoga bermanfaat.

## Referensi

- [1] <https://www.educba.com/lxc-vs-lxd/>
- <https://discuss.linuxcontainers.org/t/comparing-lxd-vs-lxc/24>
- <https://github.com/dustinkirkland/instance-type/blob/master/yaml/aws.yaml>

From:  
<https://wiki.samsul.web.id/> - **Samsul Maarif**

Permanent link:  
<https://wiki.samsul.web.id/linux/Menginstall.LXD.di.Ubuntu.20.04>

Last update: **2021/06/15 17:24**

