Container - LXC

Linux Containers (LXC) were first introduced in 2008 and are still widely used today.

Virtual Machines were being used for major deployments inside cloud providers or internal data centers in order to segment physical computing resources. Virtual Machines provide resource isolation and segmentation but are slow to start and require emulated CPU instructions to function. Although technologies like Intel VT-x and AMD-V provided solutions specifically to avoid emulation, the performance is not equal to a bare metal machine with the same specs.

Pasted from <https://logz.io/blog/docker-vs-kubernetes/>

Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.

What is LXC: The original Linux container technology
LXC is a userspace interface for the Linux kernel containment features. Through a powerful API and simple tools, it lets Linux users easily create and manage system or application containers.

Current LXC uses the following kernel features to contain processes:

- Kernel namespaces (ipc, uts, mount, pid, network and user) - Currently, Docker uses five namespaces to alter processes view of the system: **Process, Network, Mount, Hostname, Shared Memory.**
- Apparmor and SELinux profiles
- Seccomp policies
- Chroots (using pivot_root)
- Kernel capabilities
- CGroups (control groups)

What is LDX: supported by Canonical

- LXD is a next generation system container manager. It offers a user experience similar to virtual machines but using Linux containers instead.
- LXD isn't a rewrite of LXC, in fact it's building on top of LXC to provide a new, better user experience. Under the hood, LXD uses LXC through liblxc and its Go binding to create and manage the containers.
- It's basically an alternative to LXC's tools and distribution template system with the added features that come from being controllable over the network.
  LXD is written in Go, it's free software and is developed under the Apache 2 license.

Linux namespaces, originally developed by IBM, wrap a set of system resources and present them to a process to make it look like they are dedicated to that process.
Linux cgroups, originally developed by Google, govern the isolation and usage of system resources, such as CPU and memory, for a group of processes.

TYPE of Containers

https://linuxcontainers.org/lxc/getting-started/

Unprivileged Containers as a user - These are the safest containers.

1. Map uid and gid in files - /etc/subuid and /etc/subgid
2. By default, user is not allowed to create network device on the host, to change that, add below line in /etc/lxc/lxc-usernet
   <username> veth lxcbr0 10 (this means user is allowed to create 10 veth devices connected to the lxcbr0 bridge.
3. Create LXC configuration file
   - Create the ~/.config/lxc directory if it doesn't exist.
   - Copy /etc/lxc/default.conf to ~/.config/lxc/default.conf
   - Append the following two lines to it:
     - lxc.id_map = u 0 100000 65536
     - lxc.id_map = g 0 100000 65536

   Those values should match those found in /etc/subuid and /etc/subgid, the values above are those expected for the first user on a standard Ubuntu system.

   Just before you create your first container, you probably should logout and login again, or even reboot your machine to make sure

that your user is placed in the right cgroups. (This is only required if cgmanager wasn't installed on your machine prior to you installing LXC.)

4. **Create first container:** http://www.itzgeek.com/how-tos/linux/ubuntu-how-tos/setup-linux-container-with-lxc-on-ubuntu-16-04-14-04.html

```
# lxc-create -t download -n my-container
# lxc-start -n my-container -d   - to create a container
# lxc-info -n my-container       - confirm the status
# lxc-attach -n my-container    - to get shell inside the container
# lxc-stop -n my-container       - to stop
# lxc-destroy -n my-container   - to destroy
```

Unprivileged Containers as root - These are the safest containers.

You need to manually allocate a uid and gid range to root in /etc/subuid and /etc/subgid. And then set that range in /etc/lxc/default.conf using lxc.id_map entries similar to those above.

Root doesn't need network devices quota and uses the global configuration file so the other steps don't apply.

Privileged Containers - created by root and running as root.

```
# sudo lxc-create -t download -n privileged-container
```