The LXD roadmap
What to expect over the next 6 months
Meet the team
The roadmap
Virtual machines

- **USB, PCI and flexible GPU passthrough**
  Local and remote USB devices, arbitrary PCI devices and mdev/SR-IOV GPUs.

- **vTPM devices**
  Persistent virtual TPM devices (encrypted storage, HSM, attestation, …).

- **Stateful snapshots & stateful stop**
  VM hibernation and state restoration, first step towards live migration.

- **VirtioFS support**
  Much faster alternative to 9p. Automatically used by guests that support it.

- **Hotplug of network devices**
  Allow adding up to 4 additional network interfaces while the VM is running.

- **OVS/OVN offloading**
  Attach an SR-IOV NIC to a specific OVS bridge.
Containers

- **vTPM devices**
  *Persistent virtual TPM devices (HSM, attestation, …).*

- **Checkpoint restore**
  *Improve the state of stateful stop, snapshots and live-migration.*

- **CGroup handling improvements**
  *Daily testing for CGroupV1 and CGroupV2, fixes to LXD and LXCFS.*
Networking

- **Security groups and firewalling**
  Supported for OVN and managed bridges, allow restricting ingress/egress on a per network/instance basis.

- **Routed external subnets & project integration**
  Make addresses external to the network available for consumption by the network and its instances.

- **Refactoring of low-level network operations**
  Currently calling “iproute” a lot, replace with native netlink API.
Clustering & projects

- **Token based cluster join**
  Allow pre-adding a cluster member, getting a one-time token back to use during join.

- **Improved intra-cluster security**
  Switch to per-cluster certificate for internal queries, clear trust entry on removal.

- **Health and warnings API**
  Record and expose warning messages, cluster-wide clear when fixed, ideal for monitoring.

- **Project resource usage**
  Report how much of the limited resources have been consumed.

- **Improved cluster image management**
  More robust replication and less racy image refreshes.
Storage

- Interactive `lxd recover` to replace clunky `lxd import`
  Interactively re-add the pools, then discover and import what’s on them.

- Usage reporting on all storage volumes
  Currently limited to instances, add new API fields to report usage on all volumes.
- **Project and role restricted client certificates**  
  *Allows for a specific certificate to be restricted to an operator role on a subset of projects.*

- **Auto-generated REST API documentation**  
  *Modern version of our dear rest-api.md.*

- **Transition to core20 for the LXD snap**  
  *Currently on core18 (Ubuntu 18.04), will get us onto Ubuntu 20.04 as a base.*
Low-level (kernel, libraries, …)

- **VFS based filesystem shifting**
  
  *Instant container creation, easy sharing of data with the host and between containers, ...*

- **Isolated user namespaces**
  
  *User namespaces with a full 32bit uid/gid range and guaranteed to be isolated from all others.*

- **Core scheduling**
  
  *Spectre/Meltdown mitigation mechanism while getting as much performance as possible.*

- **Per PID namespace max_pid value**
  
  *Allow containers to reduce their max_pid value.*

- **CGroup + pidfd integration**
  
  *Support for killing all processes in a cgroup.*
The rest
We’re hiring!

https://grnh.se/b9bb75f21us
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