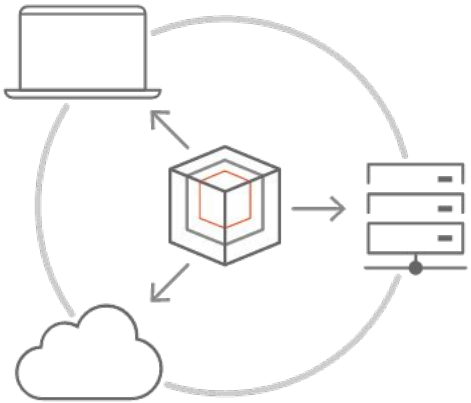
LXD

Faster, denser, lower latency Linux virtualization

CANONICAL ubuntu

LXD - your own low-touch cloud at any scale

Suitable for any environment



Runs on:

- Bare metal servers
- Workstations, laptops, Raspberry Pis…
- Virtualized platforms
- Public clouds
- Hybrid configuration - bare-metal and virtualized nodes

Secure and dependable

- Unprivileged containers by default
- Namespaces, AppArmor, Seccomp used for multi-layered security
- Advanced resource control allows for additional restrictions
- For multi-tenant environments: remote authentication, role-based access control, projects and multi-user setup

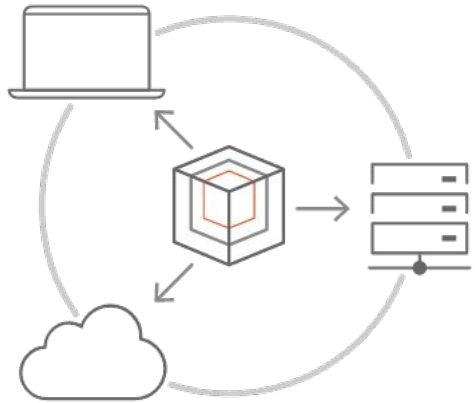
2022 — **LXD 5.0**
Small scale cloud experience

2020 — **LXD 4.0**
+VMs

2018 — **LXD 3.0**
+ Clusters

2016 — **LXD 2.0**
Production ready LTS

2008 — **LXC**
System containers



Your own low-touch cloud
at any scale

Guest images

Containers (x86_64, i686, armv7l, aarch64, ppc64le, s390x)

Virtual machines (x86_64, aarch64, ppc64le, s390x)

Host OS

Client OS

LXD runs and manages:

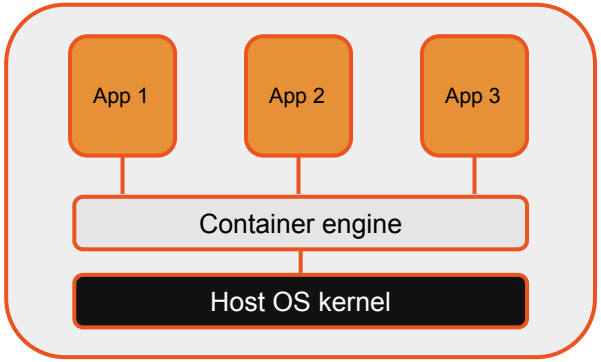
System containers

Virtual Machines

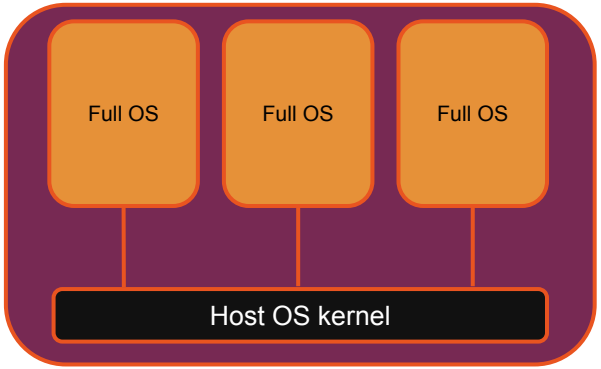
Clusters

Application containers like Docker
host a single process on a filesystem

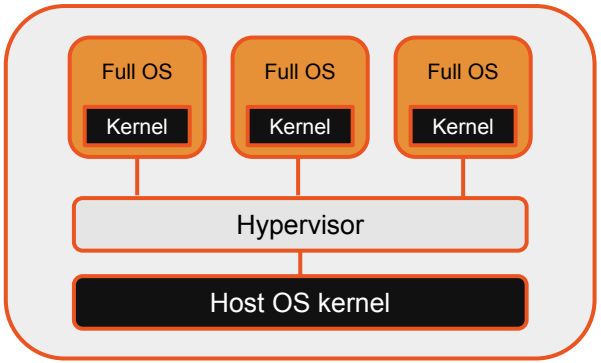
System containers from LXD
boot a full OS on their filesystem



APPLICATION CONTAINERS



SYSTEM CONTAINERS



VIRTUAL MACHINES

	Application Containers e.g. Docker, Rocket, RunC	**System Containers** e.g. LXD
What does it run?	A single running process, app, or service (e.g. /usr/sbin/mysqld)	A functional, running operating system with apps (e.g. Ubuntu, CentOS)
What does it look like inside?	Complete filesystem	Complete filesystem with background processes
What problem does it solve?	App distribution	Cheaper, faster virtual machines
Density?	Hundreds of containers per core	Hundreds of containers per core
Performance?	Identical performance to bare metal	Identical performance to bare metal
Latency?	Negligible latency	Negligible latency
How is it used at scale?	PaaS -- hosted 12-factor applications e.g. Kubernetes, Mesos	IaaS -- hosted machines e.g. OpenStack
Security?	cgroups, namespaces, (apparmor, seccomp)	cgroups, user namespaces, apparmor, seccomp

Ready in seconds

snap install lxd

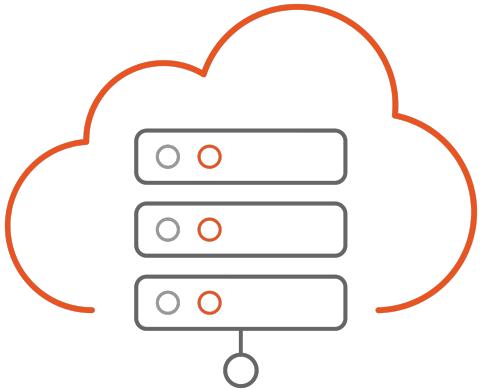
lxd init

lxc launch ubuntu:22.04 ubuntu

OR

lxc launch ubuntu:22.04 ubuntu -- vm

LXD runs and manages:
System containers
Virtual Machines
Clusters



For workloads that need a different OS or kernel

KVM/QEMU based

Modern Q35 layout with UEFI and SecureBoot by default

Architectures: x86_64, aarch64, ppc64le and s390x

Several options for launching VMs

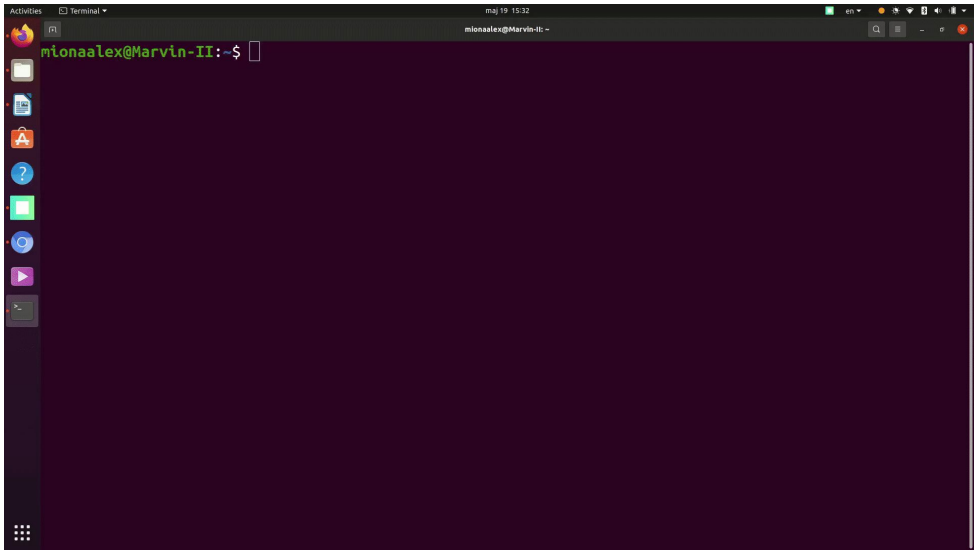
1. Any distribution from the community image server

   lxc launch ubuntu:22.04 ubuntu --vm

2. Instant desktop VM

   lxc launch images:ubuntu/22.04/desktop ubuntu --vm -c limits.cpu=4 -c limits.memory=4GiB --console=vga

3. From any ISO file - Windows included



Option 2 - instant desktop VM

LXD runs and manages:
System containers
Virtual Machines
Clusters

Up to 50 servers in a unified cluster

Mix of containers and VMs

Same distributed database

Managed uniformly

Setting up a cluster is simple

Basic cluster

- Local storage, fan overlay network
- Default settings sufficient

HA cluster

- Minimum 3 systems
- Ceph for remote storage and OVN for advanced networking

```
Would you like to use LXD clustering? (yes/no) [default=no]: yes
What IP address or DNS name should be used to reach this node? [default=192.168.0.18]:
Are you joining an existing cluster? (yes/no) [default=no]:
What name should be used to identify this node in the cluster? [default=Marvin-II]:
Setup password authentication on the cluster? (yes/no) [default=no]:
Do you want to configure a new local storage pool? (yes/no) [default=yes]:
Name of the storage backend to use (zfs, btrfs, dir, lvm) [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]:
Do you want to configure a new remote storage pool? (yes/no) [default=no]:
Would you like to connect to a MAAS server? (yes/no) [default=no]:
Would you like to configure LXD to use an existing bridge or host interface? (yes/no) [default=no]:
Would you like to create a new Fan overlay network? (yes/no) [default=yes]:
What subnet should be used as the Fan underlay? [default=auto]:
Would you like stale cached images to be updated automatically? (yes/no) [default=yes]:
Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]:
```

Unified management of your instances

- Run system containers, virtual machines or a combination of the two, all managed in the same way through the same interface
- Operate instances from any node in the cluster
- Designate a specific target node for the instance
    - Including placement based on server groups and architectures

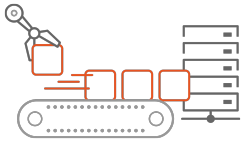
Micro Cloud

CANONICAL ubuntu

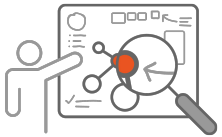
Serving compute near the consumer

Automotive

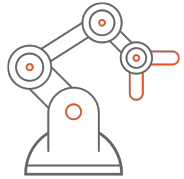
Retail



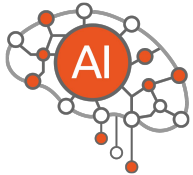
Industrial



Education and Offices



Robotics



Applied AI

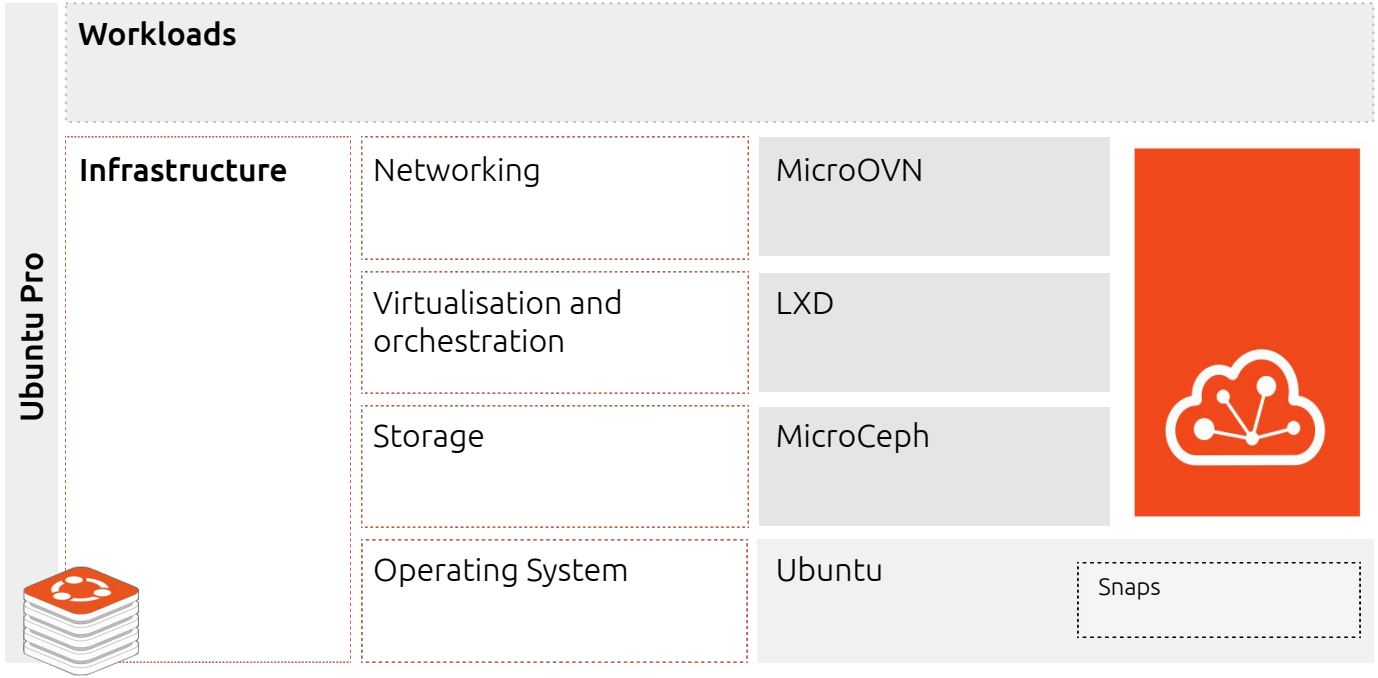
Healthcare

Telco

Gaming

3-50 servers

Public Cloud | Private Cloud | **MicroCloud** | Internet of Things



Public Cloud

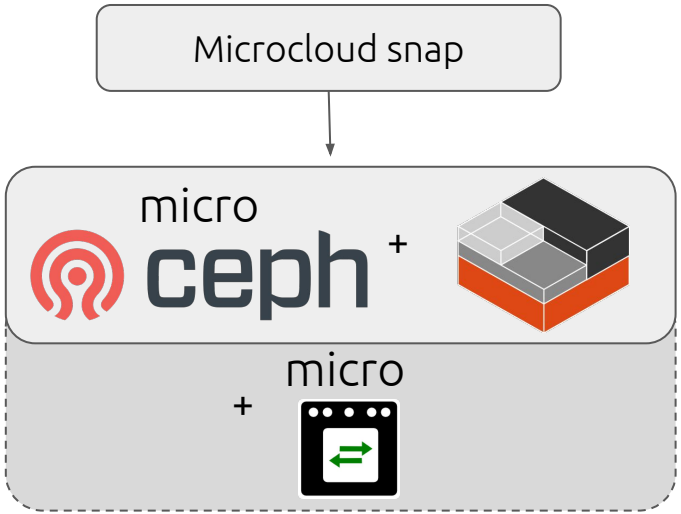
Private Cloud

MicroCloud

Internet of Things

What changed in the new approach?

- **Snap** based
  - Easy and replicable deployment every time
  - Increased security
  - Easier updates
- Can be deployed on a regular **Ubuntu Server or on Ubuntu Core** for even more lightweight solution
- Still mostly the same components (LXD, Ceph, OVN, k8s) but **tightly integrated**
  - The user doesn't need to configure each of these components, but rather just deploys via micro cloud snap

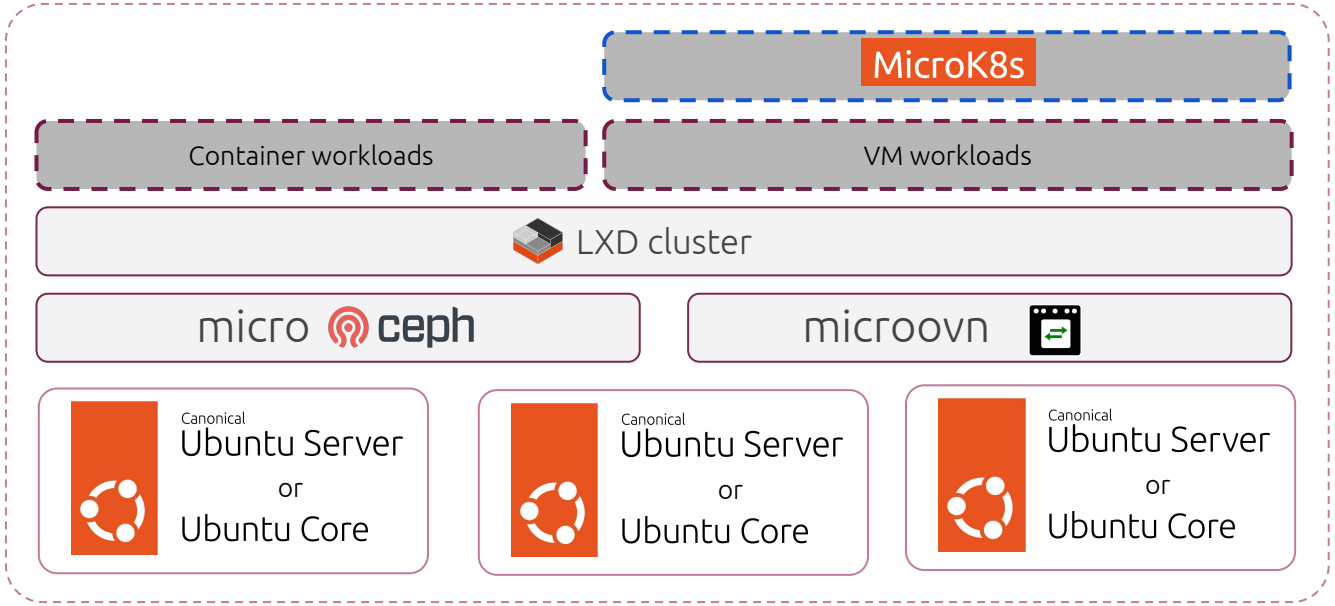


Cloud-ish

- Automation
- API access
- Diverse substrates
- Economics

but localised

- 3-10s servers (HA)
- 1000s of remote sites
- Light control plane
- Remote deploy and manage



Public Cloud

Private Cloud

Micro Cloud

Internet of Things

Snaps everywhere

snap install **lxd**

snap install **microceph**

snap install **microovn**

snap install **microcloud**

Bootstrapping

MVP flow

Ship server with appliance image on disk

Plug servers on network

Servers auto-update to latest version of all snaps

Run “microcloud bootstrap” on one server (either locally or over SSH)

Microcloud bootstrap will:

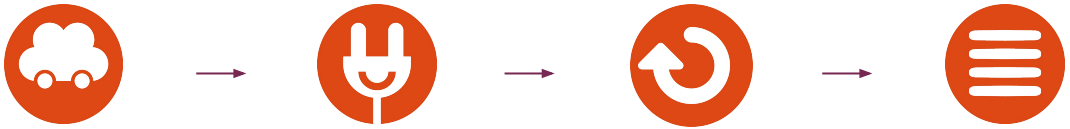
Detect all the other servers on the network and allow their immediate inclusion

Show available disks on all servers and allow adding to Ceph

Prompt about network setup (direct bridge to physical or setup some VLANs)

After bootstrap, user has a working LXD cluster connected to Ceph and their network, ready to run workloads

Target flow



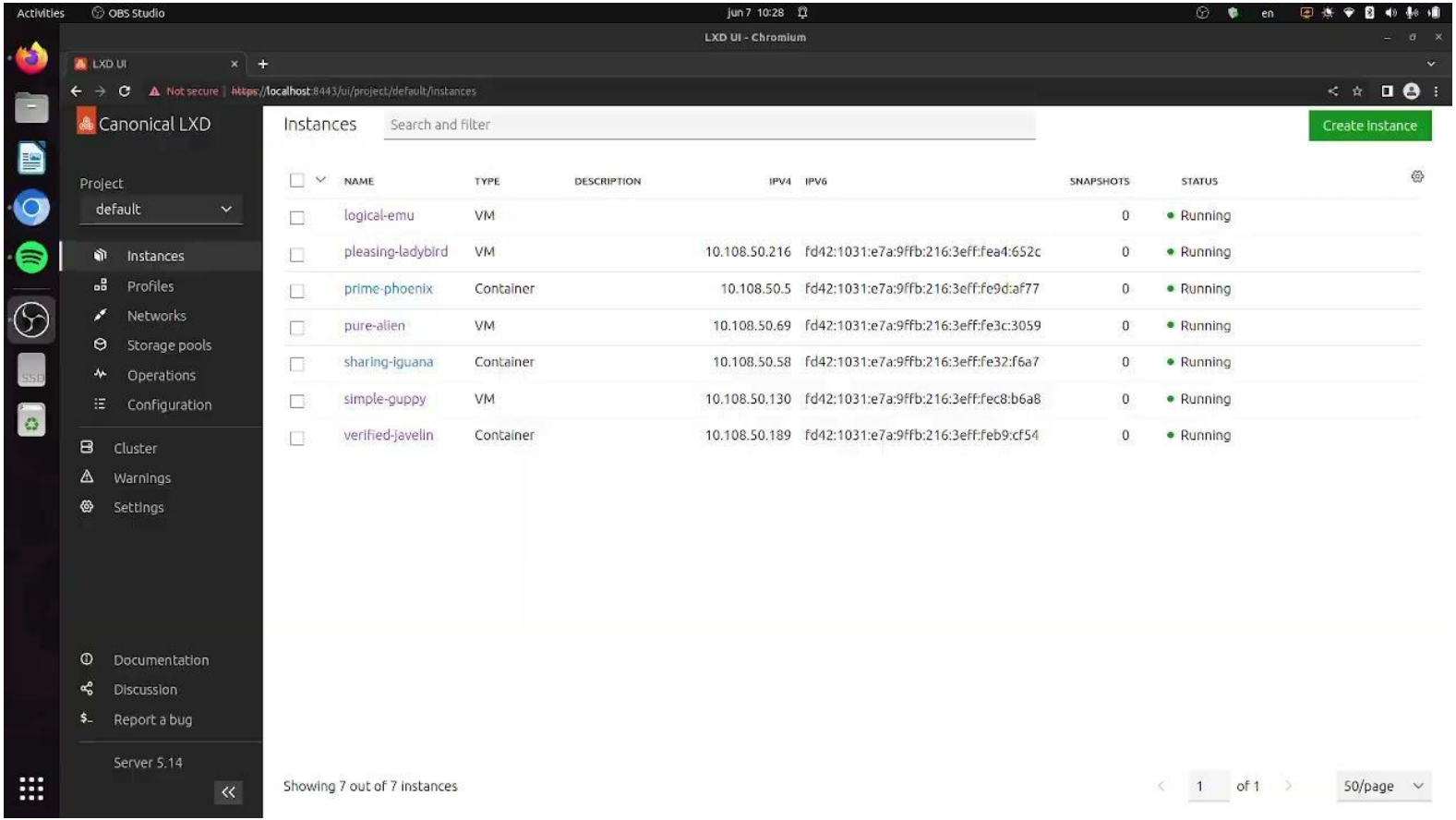
Ship server with appliance image on disk and target Central MAAS server credentials

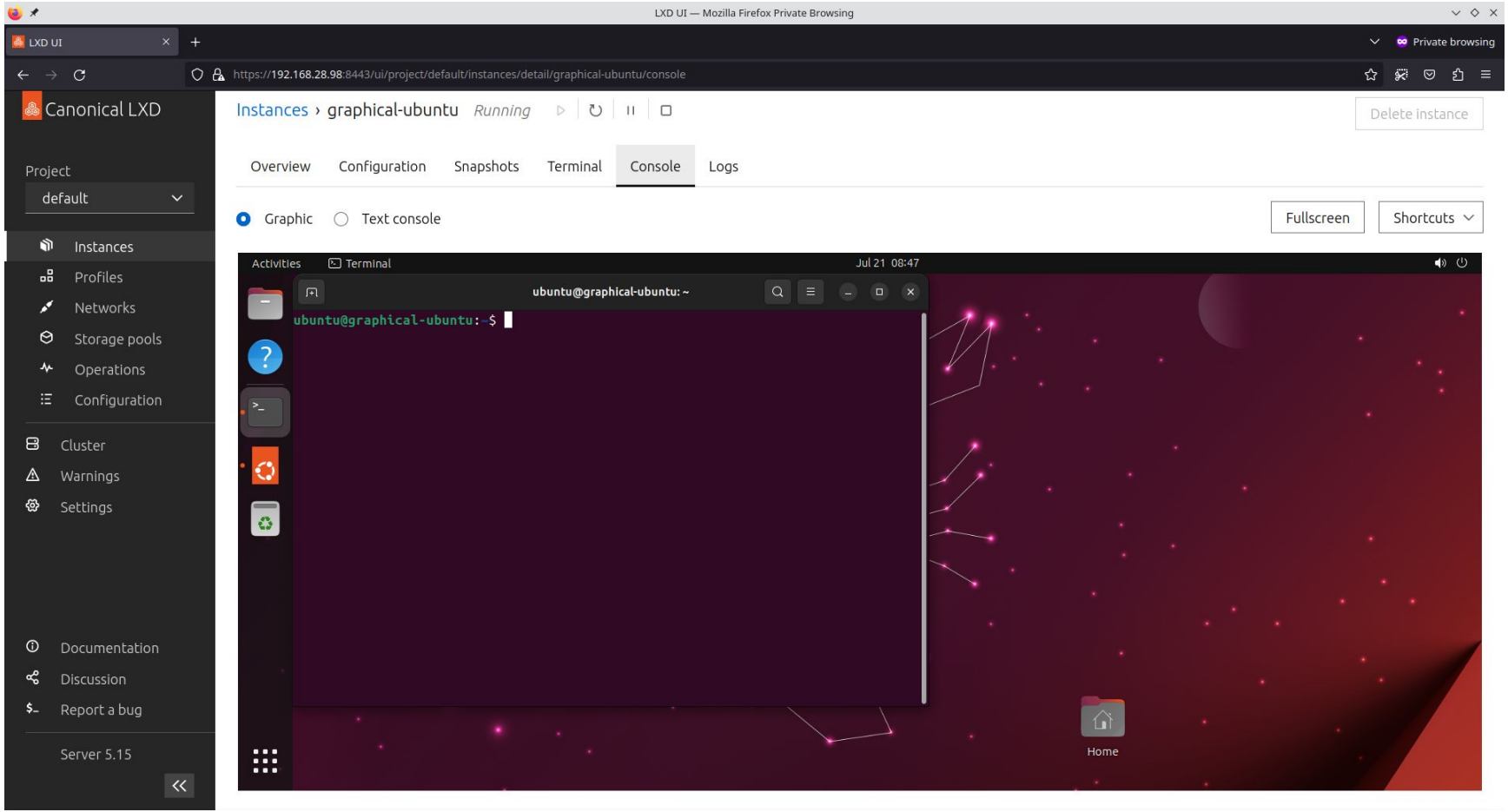
Plug servers on network

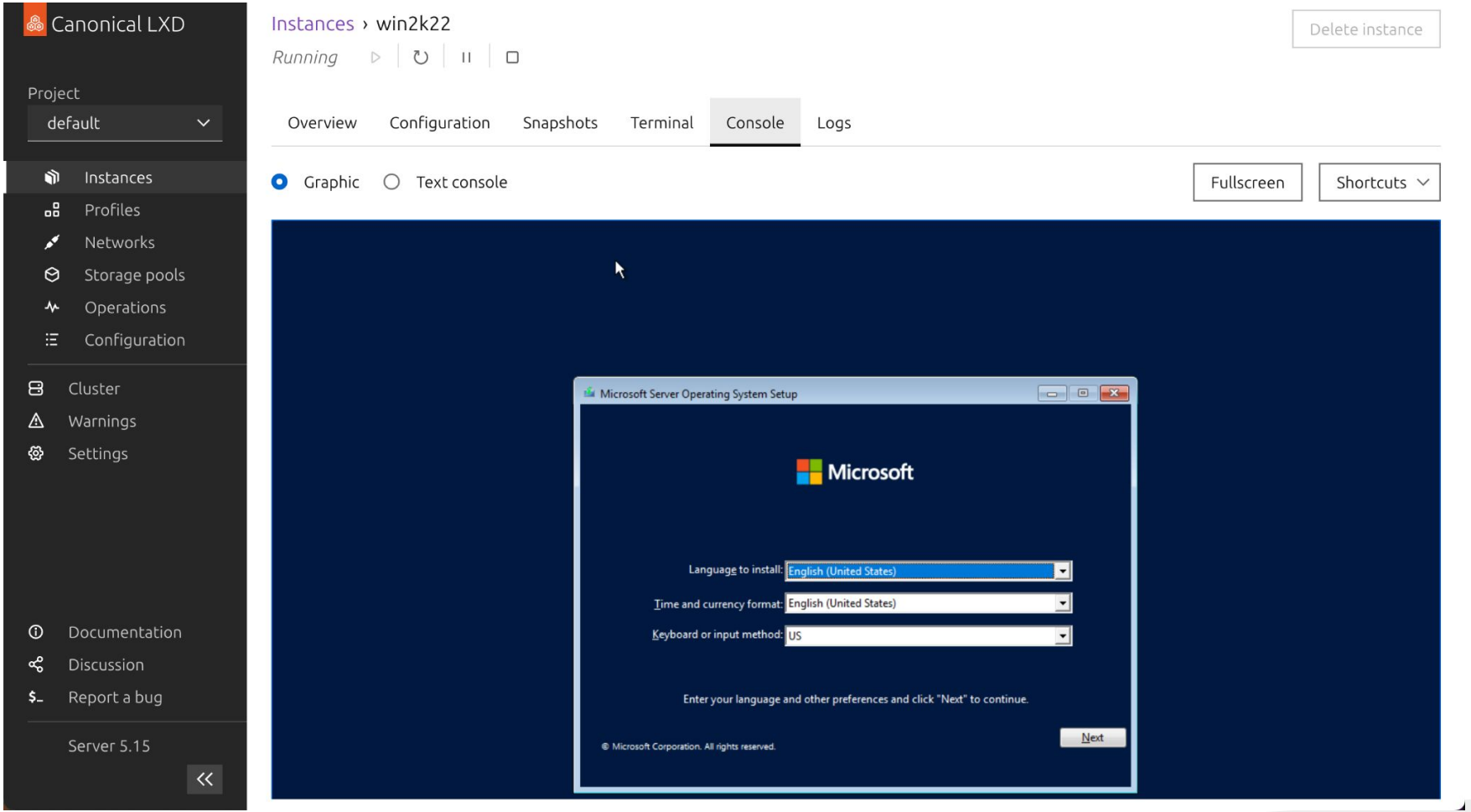
Servers auto-update to latest version of all snaps

Servers connect to the Central MAAS server, show up as a new site ready to configure, rest is done from central location

UI is in tech preview




LXD UI — Mozilla Firefox Private Browsing
LXD UI
Private browsing
https://192.168.28.98:8443/ui/project/default/instances/detail/graphical-ubuntu/console
Canonical LXD
Project
default
Instances
Profiles
Networks
Storage pools
Operations
Configuration
Cluster
Warnings
Settings
Documentation
Discussion
Report a bug
Server 5.15
Instances › graphical-ubuntu Running
Delete instance
Overview
Configuration
Snapshots
Terminal
Console
Logs
Graphic
Text console
Fullscreen
Shortcuts
Activities
Terminal
Jul 21 08:47
ubuntu@graphical-ubuntu: ~
ubuntu@graphical-ubuntu:~$
Home


Canonical LXD
Project
default
Instances
Profiles
Networks
Storage pools
Operations
Configuration
Cluster
Warnings
Settings
Documentation
Discussion
Report a bug
Server 5.15
Instances › win2k22
Running
Delete instance
Overview
Configuration
Snapshots
Terminal
Console
Logs
Graphic
Text console
Fullscreen
Shortcuts
Microsoft Server Operating System Setup
Microsoft
Language to install: English (United States)
Time and currency format: English (United States)
Keyboard or input method: US
Enter your language and other preferences and click "Next" to continue.
© Microsoft Corporation. All rights reserved.
Next

Features

Networking options supported

Fully controlled networks

- **Network bridge** (default) - a virtual L2 ethernet switch that instance NICs can connect to. The bridge can also provide local DHCP and DNS
- **OVN Network** - creates an OVN based logical network. Non-admin users can manage their own OVN network even in a restricted project

External networks

- **Macvlan network** - virtual LAN, splitting up the network interface into several sub-interfaces with their own IP addresses
- **SR-IOV network** - hardware standard allowing a single network card port to appear as several virtual network interfaces in a virtualized environment
- **Physical network** - connects to an existing physical network (a network interface or a bridge), and serves as an uplink network for OVN

Storage drivers

Supported storage drivers

- Directory - dir
- Btrfs - btrfs
- LVM - lvm
- ZFS - zfs
- Ceph - ceph
- CephFS - cephfs

Storage location	Directory	Btrfs	LVM	ZFS	Ceph	CephFS
Shared with the host	✓	✓	-	✓	-	-
Dedicated disk/partition	-	✓	✓	✓	-	-
Loop disk	✓	✓	✓	✓	-	-
Separate storage	-	-	-	-	✓	✓

Backup and export

Backup and recovery - from an instance to a full backup and disaster recovery

Snapshots - save and restore the state of an instance

Container and image transfer

Live migration - consolidated tool (lxd-migrate) for turning physical systems into containers or VMs

Migration

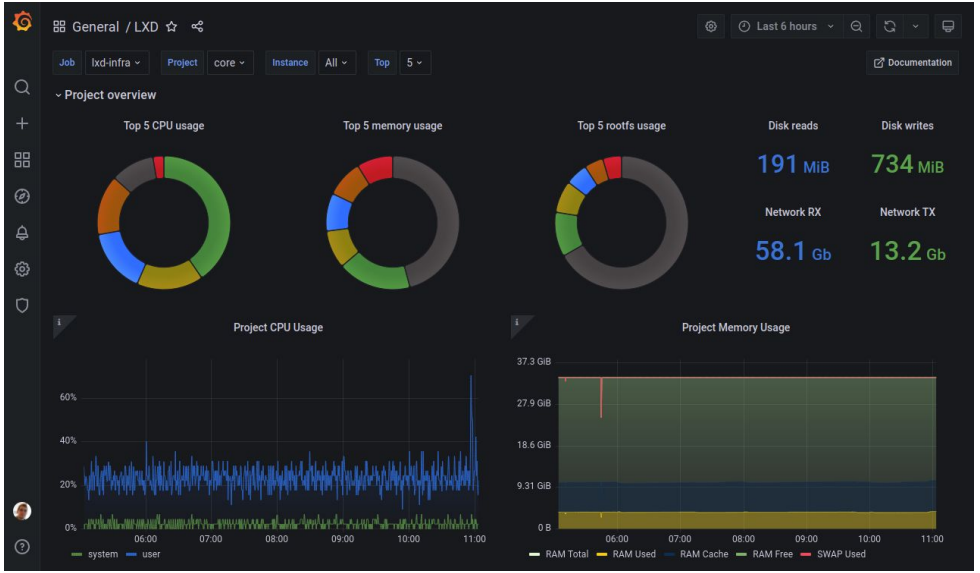
LXD migrate tool enables:

- Migration of existing LXD instances between servers
- Migration of physical or virtual machines to LXD instances
- Migration of instances from LXC to LXD

- *LXD migrate tool creates a LXD instance based on an existing disk or image*
- *Run the tool on any Linux machine*
- *It connects to a LXD server and creates a blank instance, which you can configure during or after the migration*
- *The tool then copies the data from the disk or image that you provide to the instance*

Instance metrics

- Covers CPU, memory, network, disk and process usage
- Meant to be consumed by Prometheus and likely graphed in Grafana
- In cluster environments, LXD will only return the values for instances running on the server being accessed
  - It’s expected that each cluster member will be scraped separately
- Currently guest instances only
  - Adding internal metrics for service monitoring planned for this cycle

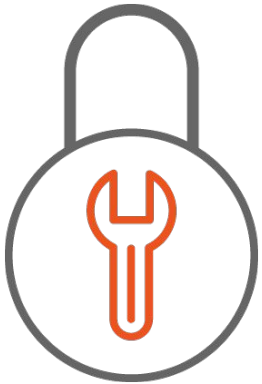


Official LXD dashboard provides per project resource consumption overview, as well as per instance data

Identity Management

Supported authentication methods:

- TLS client certificates
  - SSH-like private/public key exchange combined with an (optional) trust password that's configured on the remote server to make adding new trusted clients easier
  - Supports more complex use cases - such as restricting to specific projects
  - Good for a limited number of clients and no need to add or revoke access very often
- Candid-based authentication
  - Identity service allowing access to variety of external identity providers such as Ubuntu SSO, LDAP, Keystone, Keycloak OpenID Connect etc.
  - Doesn't allow for project or user based restrictions
- Canonical Role Based Access Control (RBAC)
  - Requires Ubuntu Advantage subscription
  - Can limit what an API client is allowed to do on LXD - authentication happens through Candid, while the RBAC service maintains roles to user/group relationships
  - Roles can be assigned to individual projects, to all projects or to the entire LXD instance.

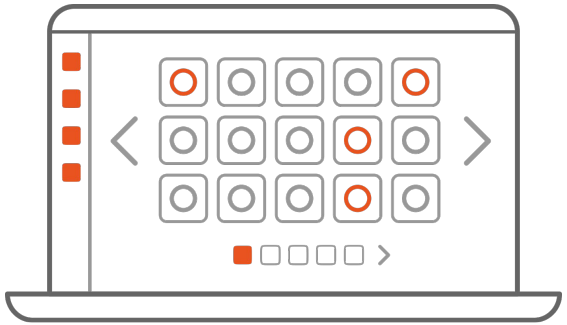


Projects and logical tenant segmentation

- Projects as a way to group your instances and/or split your LXD server
- Each project holds its own set of instances and may also have its own images and profiles
- *lxc project set <project> <key> <value>* lets users define:
  - Separate set of images and image aliases for the project
  - Separate set of networks for the project
  - Separate set of profiles for the project
  - Separate set of storage volumes for the project
  - Various image management features (such as auto update interval or default architecture…)
  - Various limits (such as number of containers/instances, CPU, disk space, memory, networks…)
  - Various restrictions (such as targeting certain cluster groups, use of low-level container options, running privileged containers, use of specific device types…)

Image management

- LXD is image based - usually downloaded from a public image server
  - Users can build images from scratch (using distrobuilder) or republish any existing instance as a new image
  - Possible to create empty instances - useful for VMs when booting from an ISO file
- LXD comes with a built-in image store where the user or external tools can import images
- Supports importing images from three different sources:
  - Remote image server (LXD or simplestreams) - supported directly at instance creation time
  - Direct pushing of the image files
  - File on a remote web server
- In clusters, LXD will replicate images on as many cluster members as you have database members



Operations in air-gapped environments: images

- In an air-gapped/firewalled environments where images cannot be directly retrieved from an external server, images can be downloaded on another system, then transferred to the target system and manually imported into the local image store
  - Create a designated image server by spinning up an empty LXD server with no storage pools and no networks
  - Copy the images
  - Set the *core.https_address* configuration option to *:8443* and do not configure any authentication methods to make the LXD server publicly available over the network on port 8443
  - Set the images that you want to share to public
  - Add images to the target system

Operations in air-gapped environments: snapd

- By default snapd will automatically update LXD to the most recent version in the user's chosen channel up to four times a day
- If the user needs absolute control over when updates are applied, the best approach is to use the Snap Store Proxy
  - Provides an on-premise edge proxy - a means to access the Snap Store for devices with restricted network access
  - Snap Store Proxy can communicate with the Snap Store directly or through an HTTPS forward proxy
  - Users can then specify that the Snap Store Proxy only makes a specific revision of LXD available to install for each architecture
- Recommended approach:
  - Set up a Snap Store Proxy
  - Run test systems on the upstream stable channels
  - Pin your proxy to the upstream revisions you've tested
  - Frequently re-evaluate and update your pinning

Security hardening and compliance

FIPS

- Requires Ubuntu Advantage
- In LXD containers, the FIPS kernel should not be installed in the instance itself, but only in the Hypervisor. As the kernel is shared from the host to the container, it will have the FIPS enabled kernel after it has been enabled in the host.
- The remaining FIPS modules, openssh server, openssh client, openssl, and strongswan may be installed into the instances and run in FIPS mode
- For LXD VMs, If the VM runs a kernel that updates the FIPS kernel, it follows the regular process

CIS

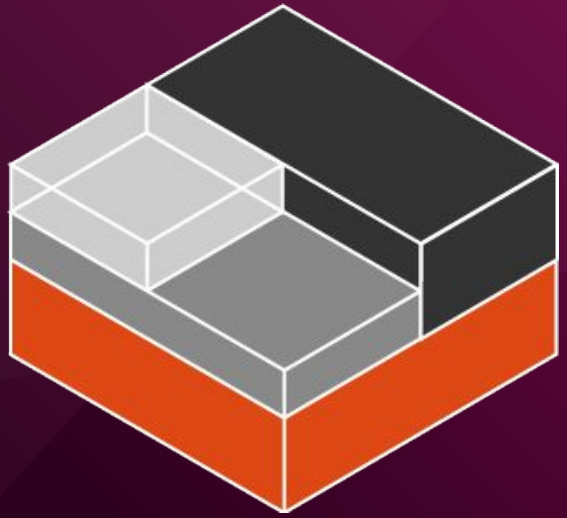
https://www.cisecurity.org/benchmark/lxd

To check what's needed

Feature comparison

Micro cloud as well has many comparable components from the perspective of what they do, but they work in a different way than those from VMware

VMware component	Comparable Micro cloud component
vSphere	LXD
ESXi	KVM/QEMU for VMs, LXC for system containers
NSX	OVN
vSAN	Ceph
vMotion	lxd-migrate
Storage vMotion	N/A
VDS	OVN
DRS	On the roadmap
vRealize Log insight	LMA
HA	HA by default
Host (anti-)affinity rules	Projects/cluster groups/scriptlet based placement
VM (anti-) affinity rules	Projects/cluster groups/scriptlet based placement
VMware Aria Automation	Not built-in -> Ansible, Terraform, Juju and other third party integrations possible



ubuntu.com/lxd

linuxcontainers.org

github.com/lxc