

# CSYE 6225: Network Structure & Cloud Computing Course

## Tutorial and Practice: VMs and Containers

### LXC

#### Objectives:

This tutorial covers basic containerization commands.

#### Prerequisites:

You will need to create a micro Ubuntu instance on AWS, ssh to the created Ubuntu virtual machine (VM), and execute this tutorial. You will install software to support these commands.

#### LXC:

LXC (Linux Containers) is an OS-level virtualization technology that allows multiple isolated Linux environments (containers) to run on a single host, sharing the same kernel but having their own file systems, processes, and network interfaces. LXC is often considered lightweight because it does not require full virtual machines (VMs) with their own kernels like traditional virtualization (e.g., KVM, VMware, etc.). Key Features of LXC are

#### LXC Components:

- **Namespaces:** Provides process isolation, network isolation, and more (e.g., PID, network, and user namespaces).
- **Control Groups (cgroups):** Allows limitation and prioritization of resources (CPU, memory, disk I/O) for containers.
- **LXC Templates:** Pre-built templates for creating containers with various Linux distributions (e.g., Ubuntu, CentOS).

#### Key Features of LXC:

1. **Lightweight:** Containers share the host system's kernel, making them less resource-intensive compared to virtual machines.
2. **Isolation:** LXC containers are isolated from each other and from the host, ensuring that processes, network interfaces, and file systems are independent.
3. **Efficiency:** Since there's no need for a full guest OS, LXC containers are more efficient in terms of memory and CPU usage.
4. **Flexible Networking:** Containers can be connected using various networking configurations (bridged, NAT, macvlan, etc.).

5. **Security:** LXC uses kernel features such as namespaces and cgroups to provide process isolation and resource limitation.

#### **How LXC Differs from Docker:**

- LXC provides a more traditional, system-level container, similar to a lightweight VM with its own init system, whereas Docker focuses on application-level containers designed for deploying specific applications or services.
- LXC allows for running a full Linux distribution, whereas Docker runs a single application or process per container.

#### **Hand-on Creating my-sql container.**

- ssh to your micro ubuntu VM on AWS.

```
$ lxc
```

```
[ubuntu@ip-172-31-31-92:~$ lxc
```

```
Installing LXD snap, please be patient.
```

```
Description:
```

```
Command line client for LXD
```

```
All of LXD's features can be driven through the various commands below.  
For help with any of those, simply call them with --help.
```

```
Usage:
```

```
lxc [command]
```

```
Available Commands:
```

alias	Manage command aliases
auth	Manage user authorization
cluster	Manage cluster members
config	Manage instance and server configuration options
console	Attach to instance consoles
copy	Copy instances within or in between LXD servers
delete	Delete instances and snapshots
exec	Execute commands in instances
export	Export instance backups
file	Manage files in instances
help	Help about any command
image	Manage images
import	Import instance backups
info	Show instance or server information
init	Create instances from images
launch	Create and start instances from images
list	List instances
monitor	Monitor a local or remote LXD server
move	Move instances within or in between LXD servers
network	Manage and attach instances to networks
operation	List, show and delete background operations
pause	Pause instances
profile	Manage profiles
project	Manage projects
publish	Publish instances as images
query	Send a raw query to LXD
rebuild	Rebuild instances
remote	Manage the list of remote servers
rename	Rename instances and snapshots
restart	Restart instances
restore	Restore instances from snapshots
snapshot	Create instance snapshots
start	Start instances
stop	Stop instances
storage	Manage storage pools and volumes
version	Show local and remote versions
warning	Manage warnings

```
Flags:
```

--all	Show less common commands
--debug	Show all debug messages
--force-local	Force using the local unix socket
-h, --help	Print help
--project	Override the source project
-q, --quiet	Don't show progress information
--sub-commands	Use with help or --help to view sub-commands

- Install LXC: Ensure LXC is installed on your system.
  - `sudo apt update`
  - `sudo apt install lxc lxc-templates`

```
ubuntu@ip-172-31-31-92:~$ sudo apt install lxc lxc-templates
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  arch-test binutils binutils-common binutils-x86-64-linux-gnu bridge-utils bzip2 cloud-image-utils
  debootstrap dns-root-data dnsmasq-base dpkg fakechroot fakeroot genisoimage libbinutils
  libboost-iostreams1.83.0 libboost-thread1.83.0 libbz2-1.0 libctf-nobfd0 libctf0 libdaxctl1
  libdistro-info-perl libdpkg-perl libfakechroot libfakeroot libfile-fcntllock-perl libgprofng0 libiscsi7
  liblxc-common liblxc1t64 libndctl6 libnfs14 libpam-cgfs libpam1 libpmem1 libpmemobj1 librados2 librbd1
  librdmacm1t64 librsync1 libsubid4 liburing2 lxcfs mmdbootstrap qemu-block-extra qemu-utils uidmap
Suggested packages:
  binutils-doc gprofng-gui ifupdown bzip2-doc mtools squid-deb-proxy-client debian-archive-keyring
  debsig-verify wodim cdrkit-doc debian-keyring gcc | c-compiler bzip2 criu python3-lxc qemu-user-static
  apt-transport-tor dpkg-dev genext2fs perl-doc qemu-user squashfs-tools-ng
The following NEW packages will be installed:
  arch-test binutils binutils-common binutils-x86-64-linux-gnu bridge-utils bzip2 cloud-image-utils
  debootstrap dns-root-data dnsmasq-base fakechroot fakeroot genisoimage libbinutils libboost-iostreams1.83.0
  libboost-thread1.83.0 libctf-nobfd0 libctf0 libdaxctl1 libdistro-info-perl libdpkg-perl libfakechroot
  libfakeroot libfile-fcntllock-perl libgprofng0 libiscsi7 liblxc-common liblxc1t64 libndctl6 libnfs14
  libpam-cgfs libpmem1 libpmemobj1 librados2 librbd1 librdmacm1t64 librsync1 libsubid4 liburing2 lxc
  lxc-templates lxcfs mmdbootstrap qemu-block-extra qemu-utils uidmap
The following packages will be upgraded:
  dpkg libbz2-1.0
2 upgraded, 46 newly installed, 0 to remove and 137 not upgraded.
Need to get 22.1 MB of archives.
After this operation, 107 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 dpkg amd64 1.22.6ubuntu6.1 [1283 kB]
Get:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 libbz2-1.0 amd64 1.0.8-5.1build0.1 [34.4 kB]
Get:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 arch-test all 0.21-1 [12.7 kB]
Get:4 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 binutils-common amd64 2.42-4ubuntu2 [239 kB]
Get:5 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 librsync1 amd64 2.42-4ubuntu2 [14.8 kB]
Get:6 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libbinutils amd64 2.42-4ubuntu2 [572 kB]
Get:7 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libctf-nobfd0 amd64 2.42-4ubuntu2 [97.1 kB]
Get:8 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libctf0 amd64 2.42-4ubuntu2 [94.5 kB]
Get:9 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libgprofng0 amd64 2.42-4ubuntu2 [851 kB]
Get:10 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 binutils-x86-64-linux-gnu amd64 2.42-4ubuntu2 [2469 kB]
Get:11 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 binutils amd64 2.42-4ubuntu2 [18.0 kB]
Get:12 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:13 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 bzip2 amd64 1.0.8-5.1build0.1 [34.5 kB]
Get:14 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dns-root-data all 2023112702-willsync1 [4450 B]
Get:15 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dnsmasq-base amd64 2.90-2build2 [375 kB]
Get:16 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 libfakechroot amd64 2.20.1+ds-15 [47.2 kB]
Get:17 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 fakechroot all 2.20.1+ds-15 [25.4 kB]
Get:18 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libfakeroot amd64 1.33-1 [32.4 kB]
Get:19 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 fakeroot amd64 1.33-1 [67.2 kB]
Get:20 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 genisoimage amd64 9:1.1.11-3.5 [378 kB]
Get:21 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libboost-iostreams1.83.0 amd64 1.83.0-2.1ubuntu3 [259 kB]
Get:22 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libboost-thread1.83.0 amd64 1.83.0-2.1ubuntu3 [276 kB]
Get:23 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libdaxctl1 amd64 77-2ubuntu2 [21.4 kB]
Get:24 http://us-west-2.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libdistro-info-perl all 1.7build1 [5616 B]
```

- **Create and Start a Container:** Create a container (using Ubuntu as an example) and start it:

`$ sudo lxc-create -t download -n mysql-container -- --dist ubuntu --release focal --arch amd64`

```
ubuntu@ip-172-31-31-92:~$ sudo lxc-create -t download -n mysql-container -- --dist ubuntu --release focal --arch amd64
Downloading the image index
Downloading the rootfs
Downloading the metadata
The image cache is now ready
Unpacking the rootfs

-----
You just created an Ubuntu focal amd64 (20240921_07:42) container.

To enable SSH, run: apt install openssh-server
No default root or user password are set by LXC.
```

`$ sudo lxc-start -n mysql-container`

```
ubuntu@ip-172-31-31-92:~$ sudo lxc-start -n mysql-container
```

**Set Up Networking (Optional):** Configure networking if needed (bridged or NAT). You can check the container's IP:

`sudo lxc-ls -f`

**Access the Container:** Log into the container:

`sudo lxc-attach -n mysql-container`

Once inside the container, update the package list:

`apt update`

```
ubuntu@ip-172-31-31-92:~$ sudo lxc-attach -n mysql-container
root@mysql-container:/home/ubuntu# apt update
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Hit:3 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:4 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [3601 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1232 kB]
Fetched 4962 kB in 2s (2282 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
```

**Install MySQL Server:** Install MySQL within the container:

`apt install mysql-server`

```

root@mysql-container:/home/ubuntu# apt install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libssl1 liblog-fast-perl liblog-pm-perl libncode-locale-perl libevent-core-2.1-7 libevent-threads-2.1-7 libfcgi-perl libgdbm-compat4 libgdbm6 libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl
  libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libnuma1 libperl5.30 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-util mysql-client-8.0 mysql-client-core-8.0 mysql-common
  mysql-server-8.0 mysql-server-core-8.0 perl perl-modules-5.30 psmisc
Suggested packages:
  gdbm-118n libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx tinycsa perl-doc libterm-readline-gnu-perl | libterm-readline-perl-perl make libdb-debug-perl liblocale-codes-perl
The following NEW packages will be installed:
  libssl1 liblog-fast-perl liblog-pm-perl libncode-locale-perl libevent-core-2.1-7 libevent-threads-2.1-7 libfcgi-perl libgdbm-compat4 libgdbm6 libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl
  libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libnuma1 libperl5.30 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-util mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server
  mysql-server-8.0 mysql-server-core-8.0 perl perl-modules-5.30 psmisc
0 upgraded, 33 newly installed, 0 to remove and 0 not upgraded.
Need to get 43.9 MB of archives.
After this operation, 365 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 mysql-common all 5.8+1.0.Subuntu2 [7496 B]
Get:2 http://archive.ubuntu.com/ubuntu focal/main amd64 mysql-client-core-8.0 amd64 8.0.30-0ubuntu0.20.04.1 [5882 kB]

```

**Configure MySQL (Optional):** You may want to adjust some MySQL settings depending on your container environment. For instance, you might bind MySQL to specific IPs by editing the MySQL configuration file:

```
nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

Set the bind address to 0.0.0.0 or your specific container's IP:

```
bind-address = 0.0.0.0
```

**Start MySQL:** Start MySQL inside the container:

```
systemctl start mysql
```

```

root@mysql-container:/home/ubuntu# systemctl start mysql

```

Set Root Password and Secure Installation: Run the secure installation script:

```
mysql_secure_installation
```

```

[root@mysql-container:/home/ubuntu# mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

[Press y|Y for Yes, any other key for No: Y

There are three levels of password validation policy:

LOW      Length >= 8
MEDIUM  Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary      file

[Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0

```

## Access mysql:

```

root@mysql-container:/home/ubuntu# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.39-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```

```

[mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.01 sec)

```

```

[mysql> show tables;
+-----+
| Tables_in_sys |
+-----+
| host_summary |
| host_summary_by_file_io |
| host_summary_by_file_io_type |
| host_summary_by_stages |
| host_summary_by_statement_latency |
| host_summary_by_statement_type |
| innodb_buffer_stats_by_schema |
| innodb_buffer_stats_by_table |
| innodb_lock_waits |
| io_by_thread_by_latency |
| io_global_by_file_by_bytes |
| io_global_by_file_by_latency |
| io_global_by_wait_by_bytes |
| io_global_by_wait_by_latency |
| latest_file_io |
| memory_by_host_by_current_bytes |
| memory_by_thread_by_current_bytes |
| memory_by_user_by_current_bytes |
| memory_global_by_current_bytes |
| memory_global_total |
| metrics |
| processlist |
| ps_check_lost_instrumentation |
| schema_auto_increment_columns |
| schema_index_statistics |
| schema_object_overview |
| schema_redundant_indexes |
| schema_table_lock_waits |
| schema_table_statistics |
| schema_table_statistics_with_buffer |
| schema_tables_with_full_table_scans |
| schema_unused_indexes |
| session |
| session_ssl_status |
| statement_analysis |
| statements_with_errors_or_warnings |
| statements_with_full_table_scans |
| statements_with_runtimes_in_95th_percentile |
| statements_with_sorting |
| statements_with_temp_tables |
| sys_config |
| user_summary |
| user_summary_by_file_io |
| user_summary_by_file_io_type |
| user_summary_by_stages

```

**Check on available services in the created container:**

```
root@mysql-container:/home/ubuntu# systemctl status
```



```
root@mysql-container:/home/ubuntu# systemctl status
● mysql-container
  State: running
    Jobs: 0 queued
  Failed: 0 units
  Since: Sun 2024-09-22 00:56:59 UTC; 24min ago
  CGroup: /
          └─.lxc
              ├─1605 /bin/bash
              ├─1615 systemctl status
              └─1616 pager
          └─init.scope
              └─1 /sbin/init
          └─system.slice
              ├─systemd-networkd.service
              │   └─129 /lib/systemd/systemd-networkd
              ├─cron.service
              │   └─132 /usr/sbin/cron -f
              ├─networkd-dispatcher.service
              │   └─135 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers
              ├─systemd-journald.service
              │   └─101 /lib/systemd/systemd-journald
              ├─mysql.service
              │   └─1447 /usr/sbin/mysqld
              ├─rsyslog.service
              │   └─136 /usr/sbin/rsyslogd -n -iNONE
              ├─console-getty.service
              │   └─144 /sbin/agetty -o -p -- \u --noclear --keep-baud console 115200,38400,9600 vt220
              ├─systemd-resolved.service
              │   └─141 /lib/systemd/systemd-resolved
              ├─dbus.service
              │   └─133 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
              └─systemd-logind.service
                  └─137 /lib/systemd/systemd-logind
root@mysql-container:/home/ubuntu#
```

**End Tutorial**