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.

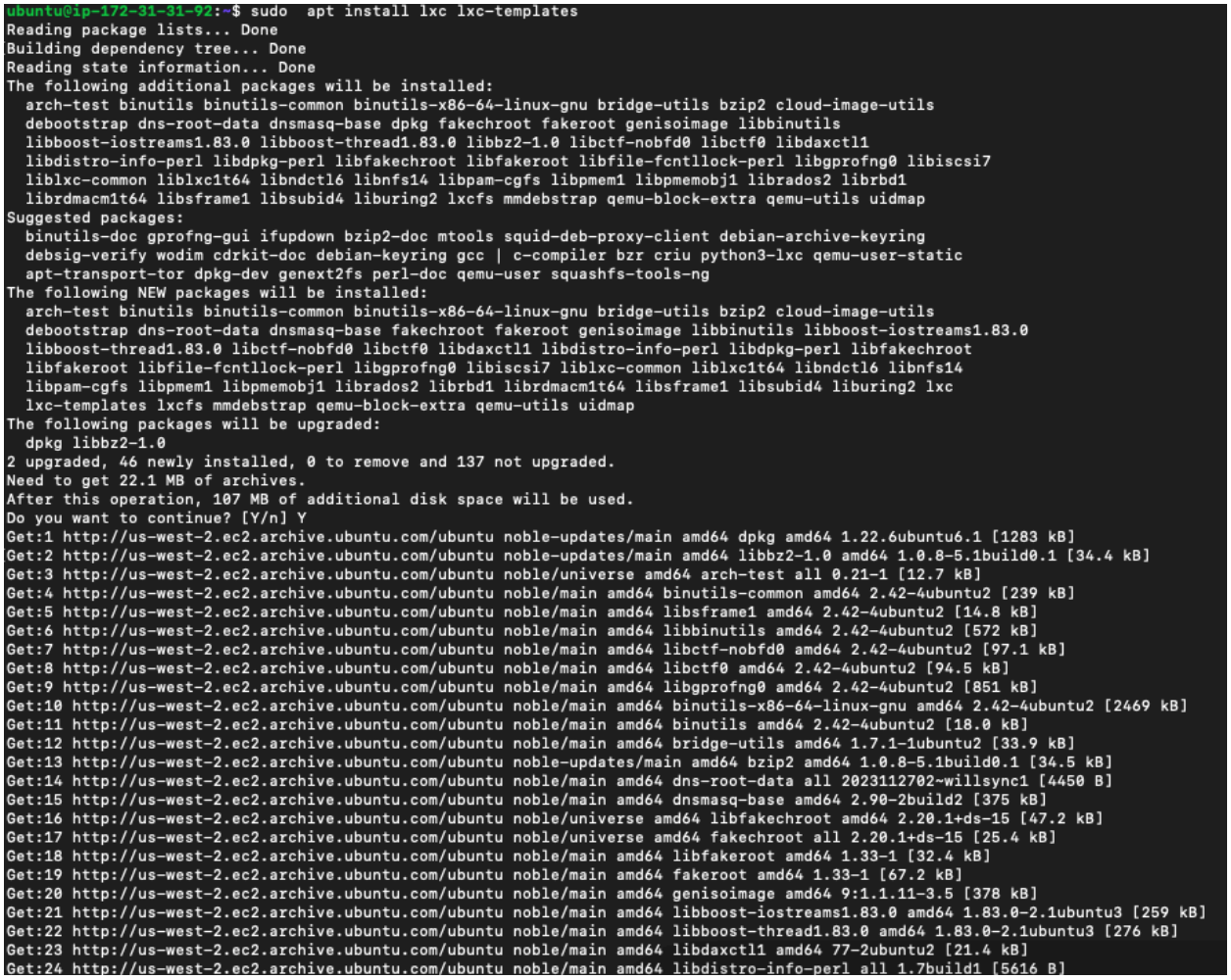
Hand-on Creating my-sql container.

- 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



- **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

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

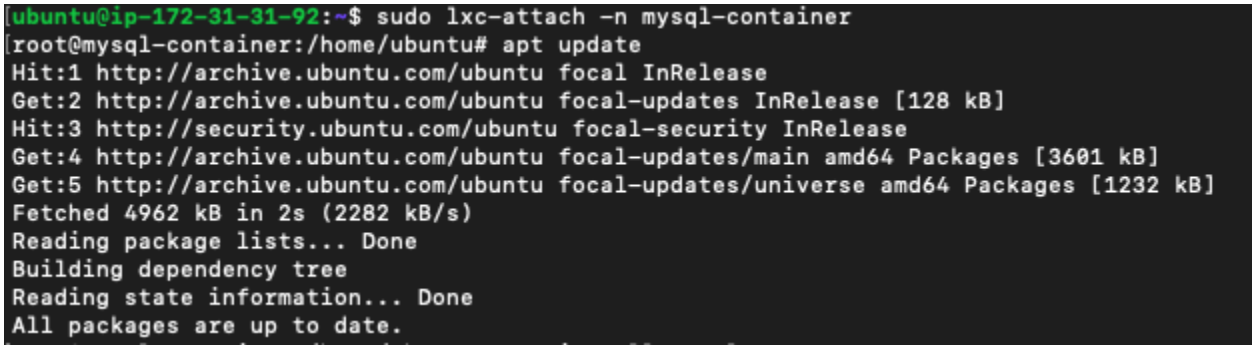
sudo lxc-ls -f

Access the Container: Log into the container:

sudo lxc-attach -n mysql-container

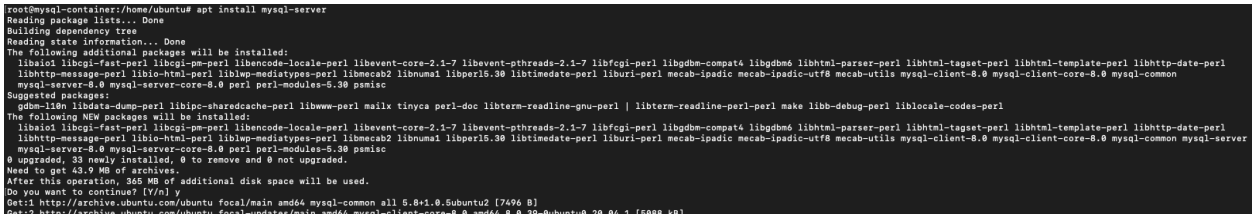
Once inside the container, update the package list:

apt update



Install MySQL Server: Install MySQL within the container:

apt install mysql-server



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:

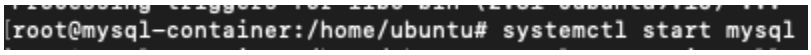
```
nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

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
```



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
```

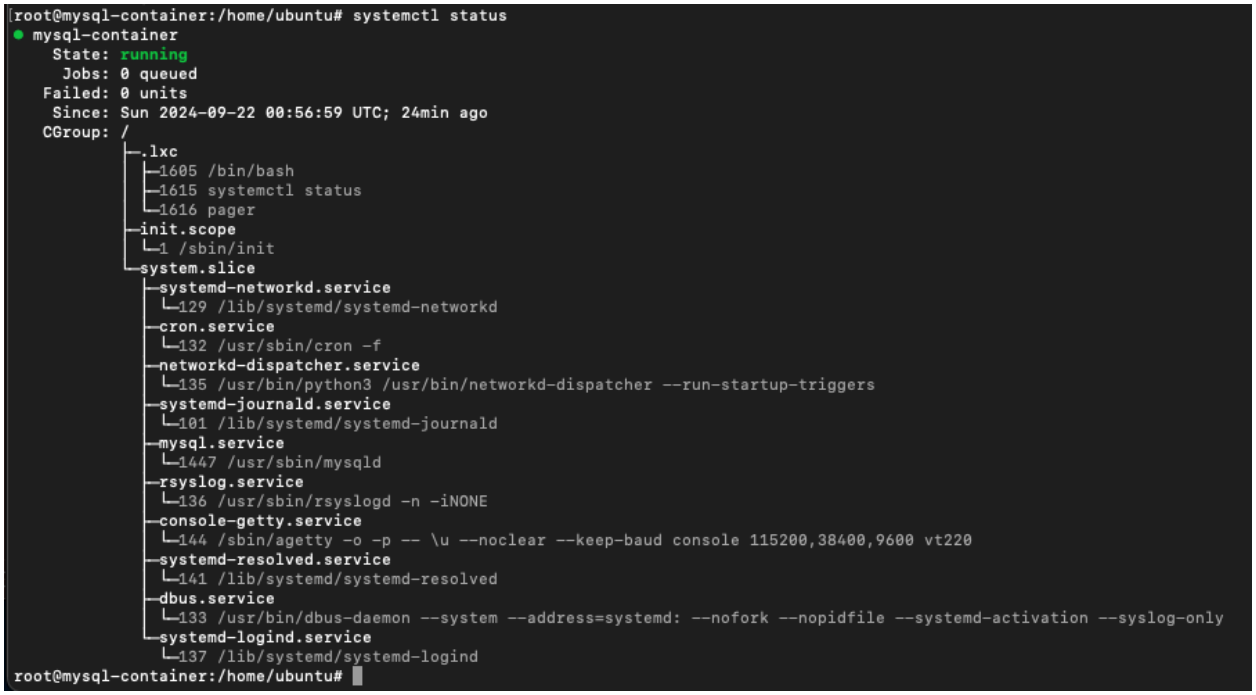
Access mysql:

```
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)
```

Check on available services in the created container:

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



End Tutorial