Running Containers with LXC
IN720 Virtualisation

Introduction

Capabilities of modern Linux kernels allow us to create *containers* that give us a lightweight and simple way to virtualise systems. We need user space tools to allow us to access these capabilities. *LXC* is a tool that allows us to create, run, and manage containers.

In this lab we will use LXC on Ubuntu servers to create and experiment with some containers.

1 Setup

A virtual Ubuntu 14.04 server has been created for you on vCloud. Use PuTTY or another ssh client to log into your server. The username is `user` and the password is `P@ssw0rd`. A list of IP addresses will be posted on the class Slack channel. We need to apply updates to our server and then install LXC, but because the servers don't seem to get a DNS server address, we must jump through a couple of hoops.

1. Edit the file `/etc/resolv.conf`, adding a line that says
   `nameserver 10.50.1.80`.
2. Apply updates with the commands
   ```
   sudo apt-get update
   sudo apt-get dist-upgrade
   ```
3. Reboot your server with the command `sudo reboot`.
4. When the system comes back up, log in again and add the nameserver line back into `/etc/resolv.conf`.
5. Now install LXC with the command `sudo apt-get install lxc`.

2 Creating New Containers

Creating a container is easy. Just enter

```
sudo lxc-create -t ubuntu -n container-01
```

This creates a new container named `container-01` based on the latest Ubuntu template. This will take a few minutes to complete as it downloads the needed packages to create the container. When this is done, create a second Ubuntu container called `container-02` and you will see that this goes much more quickly.

Even though the host system is running Ubuntu 14.04, the containers we created are running the latest Ubuntu version (It is possible to specify a particular version.) We aren't limited to Ubuntu containers, however. Create a third container called `container-03` using the Debian template.

To get a list of the containers on a host, use the command

```
sudo lxc-list
```

3 Running Your Containers

to use a container you will need to start it. Then you cna attach a console to it to work within the container. Start your `container-01` container with the command

```
sudo lxc-start -d -n container-01
```

The `-d` tells LXC to start the container without attaching a console to it automatically, which is generally what we want.

To get a console session on our container, we use the `lxc-console` command as follows:

```
sudo lxc-console -n container-01
```

Within the container, we can make changes, install software, and run services. Use the `ifconfig` command to note our container's IP address. Containers get a private IP address, but it is possible to route traffic through the host system to our container so that it can provide network services.

Hit `Ctrl-a q` to disconnect from the console.

4 Freezing and Stopping Containers

We can pause a running container with the command

```
sudo lxc-freeze -n <container-name>
```

and later resume it with the corresponding `lxc-unfreeze` command.

We can stop a container with the command

```
lxc-stop -n <container-name>
```

5 Container Storage

Containers data is stored by default in `/var/lib/lxc`. Each container has a directory containng its information. A stopped container can be cloned with the command

```
sudo lxc-clone -o <original-container-name> -n <new container>
```

6 Conclusion

LXC containers are useful and powerful, but some additional management tools will make them even more so. Next time we will start working with Docker, which builds on LXC.