Introduction to Unprivileged Container (LXC)

Topics Discussed

1. Install LXC on ubuntu 20.04
2. Environment setup for LXC
3. create,start,attach and destroy a container
4. Create a bridge for LXC container
5. IP configuration for LXC containers
6. Backup containers
7. Restore containers

Install LXC:

For installing LXC type following command:

```
sudo apt-get install lxc
```

LXC creates a NATed bridge called lxcbr0 for you. Each container will have one veth NIC and all traffic routed using the lxcbr0 bridge. To view current settings, enter:

```
sudo brctl show
```

To see an IP address assigned to the lxcbr0, enter:

```
sudo ifconfig lxcbr0
```

Environment Setup:

Step 1: Create new user (Without sudo privilege)

```
sudo adduser <user_name>
```

Step 2: Find out allocated subuids and subgids for the LXC user

```
sudo grep <user_name> /etc/sub{uid,gid}
```

To see uid and gid

```
id <user_name>
```

Step 3: Create a default container configuration file for lxc user

```
sudo nano /etc/lxc/lxc-usernet
```

Append the following line:

```
user_name veth lxcbr0 10
```

Step 4: Switch to newly created user

```
su – user_name (Best Practise logout and login)
```

Step 5: Make Dir for LXC config

```
mkdir -p ~/.config/lxc
```

Step 6: copy default configuration

```
cp /etc/lxc/default.conf ~/.config/lxc/default.conf
```

Step 7: Edit configuration file

```
nano ~/.config/lxc/default.conf
```

Append following line

```
lxc.idmap = u 0 100000 65536
lxc.idmap = g 0 100000 65536
```

Create, Start Attach and Destroy a container

Create a container:

```
lxc-create -t download -n <container_name> -- --no-validate
```

Specify all specs:

```
lxc-create -t download -n <container_name> -- -d ubuntu -r focal -a amd64 --no-validate
```

[--no-validate]: **Disable GPG validation**

List of containers:

```
lxc-ls -f
```

Start a container:

```
lxc-start -n c1
```

Access a container

```
lxc-attach -n c1
```

Stop a container

```
lxc-stop
```

destroy a container

```
lxc-destroy
```

See ubuntu version

```
lsb_release -a
```

reference: *https://www.cyberciti.biz/faq/how-to-create-unprivileged-linux-containers-on-ubuntu-linux/*

Create a bridge for LXC container Ubuntu 20.04

Step 1: Initialize Ethernet interface

To see IP address

```
ip a
```

Ubuntu 20.04 uses netplan as a default network manager. The configuration file for the netplan is stored in the /etc/netplan directory

```
ls /etc/netplan
```

Before making any changes to this file, make sure to create a backup copy of it. Use the cp command to do so:

```
sudo cp /etc/netplan/01-network-manager-all.yaml 01-network-  manager-all.yaml.bak
```

Here we are using the Nano text editor for editing yml.

```
sudo nano /etc/netplan/01-network-manager-all.yaml
```

Yaml File:

```
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    enp3s0:
      dhcp4: yes
      addresses: []
```

Then validate yaml through following command

```
sudo netplan try
```

Then apply yaml through following command

```
sudo netplan apply
```

reference: https://linuxhint.com/ubuntu_

Step 2: Create a bridge ans set a static IP for bridge

Here we are using the Nano text editor for editing yml.

```
sudo nano /etc/netplan/01-network-manager-all.yaml
```

Yaml file:

```
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    enp3s0:
      dhcp4: yes
      addresses: []
  bridges:
    br0:
      dhcp4: no
      interfaces:
       - enp3s0
      addresses:
        - 192.168.2.140/24
      gateway4: 192.168.2.254
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
```

Then validate yaml through following command

```
sudo netplan try
```

Then apply yaml through following command

```
sudo netplan apply
```

Then use the brctl command to show all bridges on the system. In this case, the Ethernet interface is automatically added as a port to the bridge.

```
sudo brctl show
```

Up an interface

```
sudo ip link set br0 up
```

Down an interface

```
sudo ip link set br0 down
```

Delete an interface

```
sudo brctl delbr br0
```

reference: *https://www.tecmint.com/create-network-bridge-in-ubuntu/*

Step 3: container Configuration

Edit Configuration for lxc network configuration

```
sudo nano /etc/lxc/lxc-usernet
```

Append the following line:

```
username veth br0 10
```

Edit container configuration

```
nano ~/.local/share/lxc/<container_name>/config
```

Set br0 at container configuration

```
lxc.net.0.link = br0
```

You can also change default configuration for every container

```
nano ~/.config/lxc/default.conf
```

For ubuntu 20.04 containers

Edit the file etc/netplan/___.yml

```
nano etcnetplan/___.yml
```

Yaml file

```
network:
  version: 2
  renderer: networkd
  ethernets:
    eth0:
      dhcp4: no
      addresses:
        - 192.168.2.148/24
      gateway4: 192.168.2.254
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
```

For ubuntu 16.04 containers

Edit the file etc/network/interfaces

```
sudo nano etc/network/interfaces
```

Interface file

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
  address 192.168.2.149
```

```
netmask 255.255.255.0
gateway 192.168.2.254
```

Backup containers:

Assume we want to backup and restore a container named c1
cd to directory ~/.local/share/lxc/

```
cd ~/.local/share/lxc/

sudo tar --numeric-owner -czvf c1.tar.gz c1
```

Restore Container

copy cantainer backup to local lxc directory

```
rsync -avhP user@ip:~/.local/share/lxc/c1.tar.gz    ~/.local/share/lxc/c1.tar.gz
```

Then extract it

```
sudo tar --numeric-owner -xzvf c1.tar.gz
```

For restoring a container we have some observation

The access permission of container is

container dir subuid:gid
config file uid:gid
rootfs subuid:subgid

then under rootfs every permission is bind by subuid and subgid

So, if our user subuid and subgid remains same in machine where we restore the container then we have to change only the permission of container dir and config file then it works fine

1. permission of container directory
2. permission of config file
3. edit mount point of config file

```
sudo chown subuid:gid container_dir
sudo chown uid:gid config
```

Edit following line in container config

```
nano ~/.local/share/lxc/c1/config
lxc.rootfs.path = dir:/home/zahid/.local/share/lxc/c1/rootfs
```

reference: *https://unix.stackexchange.com/questions/397092/what-do-the-contents-of-etc-subuid-mean-in-the-context-of-docker*

Future Task to Do: Make a script that converts all file permission under rootfs by giving new subuid (subuid and subgid remains same)