

服务器LXD虚拟化

ubuntu server版: <http://cdimage.ubuntu.com/releases/>

教程地址: https://github.com/shenuiuin/LXD_GPU_SERVER

宿主机的安装与配置

这里演示使用ubuntu-18.04.1-server-amd64.iso
使用虚拟机来演示如何安装与配置

服务器一般有一块SSD和多块机械做成的RAID的阵列，
系统安装在SSD（比较小）还有一块RAID阵列的数据盘

这里演示使用ubuntu虚拟机来演示
系统有两块硬盘，一块40G，作为系统盘，另一块
160G作为数据盘，虚拟机的安装就不介绍了

```
icon@ubuntu:~$ sudo fdisk -l
[sudo] password for icon:
Disk /dev/sda: 40 GiB, 42949672960 bytes, 83886080 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x29aa4df1

Device      Boot Start      End  Sectors  Size Id Type
/dev/sda1   *    2048 83884031 83881984   40G 83 Linux

Disk /dev/sdb: 160 GiB, 171798691840 bytes, 335544320 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
icon@ubuntu:~$ _
```

宿主机准备：作为GPU服务器，宿主机的显卡驱动等
都需要安装好，参考下面链接安装NVIDIA驱动、
CUDA、cuDNN

（虚拟机中没有英伟达的显卡，这里演示略过）

<https://medium.com/@cjanze/how-to-install-tensorflow-with-gpu-support-on-ubuntu-18-04-lts-with-cuda-10-nvidia-gpu-312a693744b5>

（需要外网才能访问，在我github上已经离线好）

宿主机准备好后，查看各个硬件是否正常工作

neofetch

显示Linux系统信息

htop

查看CPU运行以及内存占用情况

nvidia-smi

查看显卡运行情况

lxd介绍

因为服务器只有一台，如果每个人都直接连接宿主机文件会变得很乱，还有一些人会安装各种不相关软件，甚至运行危害服务器的命令



因此，虚拟化迫在眉睫
以下是我们要达到的目的：

需要将容器保存在宿主机的数据盘中
每一个单独的容器就是一个单独的系统
不同用户之间不能相互影响且可以同时使用
用户可以方便地访问自己的“机器”（远程连接）
用户有系统所有的权限，能自由地安装程序
用户不被允许直接操作宿主机
系统：cpu、内存、显卡，都可以配置

LXD/LXC是一个系统容器。Docker是一个应用程序容器。而VirtualBox不能通显卡。所以我们采用LXD虚拟化

lxd安装

安装LXD:

分别安装LXD， ZFS和bridge-utils

LXD 实现虚拟容器

ZFS 用于管理物理磁盘，支持LXD高级功能

bridge-utils 用于搭建网桥

`sudo apt-get install lxd zfsutils-linux bridge-utils`

配置网桥:

因为学校信息中心网络问题，如果配置桥接网卡，会导致流量异常，直接断网，因此实现每人一个ip的方式失败，不得已我们采用端口转发的方式来实现各个容器的网络

配置ZFS

首先，我们运行`sudo fdisk -l`列出服务器上的可用磁盘和分区，我们有两块硬盘，第一块为系统盘，第二块为数据盘，现在我们将数据盘（/dev/sdb）分出需要使用的空间，作为容器的存储卷。

`sudo fdisk /dev/sdb`

```
icon@ubuntu:~$ sudo fdisk /dev/sdb
Welcome to fdisk (util-linux 2.31.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x640c32a1.

Command (m for help):
```

按照下图分出了80GB的分区作为容器的存储卷，分区为/dev/sdb1，剩下的空间同理可以分区，可以作为服务器另外的应用使用

```
Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-335544319, default 2048): 2048
Last sector, +sectors or +size{K,M,G,T,P} (2048-335544319, default 335544319): +80G

Created a new partition 1 of type 'Linux' and of size 80 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

LXD配置

在块设备 /dev/sdb1 上创建一个ZFS存储池

```
sudo lxc storage create zfs-pool zfs source=/dev/sdb1
```

LXD初始化

```
sudo lxd init
```

```
icon@ubuntu:~$ lxd init
Would you like to use LXD clustering? (yes/no) [default=no]: no
Do you want to configure a new storage pool? (yes/no) [default=yes]: no
Would you like to connect to a MAAS server? (yes/no) [default=no]: no
Would you like to create a new local network bridge? (yes/no) [default=yes]: yes
What should the new bridge be called? [default=lxdbro]: lxdbro
What IPv4 address should be used? (CIDR subnet notation, "auto" or "none") [default=auto]: auto
What IPv6 address should be used? (CIDR subnet notation, "auto" or "none") [default=auto]: auto
Would you like LXD to be available over the network? (yes/no) [default=no]: no
Would you like stale cached images to be updated automatically? (yes/no) [default=yes]: yes
Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]: yes
config: {}
networks:
- config: {}
  ipv4.address: auto
  ipv6.address: auto
  description: ""
  managed: false
  name: lxdbro
  type: ""
storage_pools: []
profiles:
- config: {}
  description: ""
  devices:
    eth0:
      name: eth0
      nictype: bridged
      parent: lxdbro
      type: nic
    name: default
  cluster: null
```

因为我们已经创建好了一个叫zfs-pool的存储池，所以在lxd初始化时不需要创建新的储存池，之后在进行配置即可

```
sudo lxc profile edit default
```

在配置时还将每个容器的硬盘大小限制为固定大小（如果没有设置，容器里面的磁盘大小为整个储存池的大小）

```
###
### Note that the name is shown but cannot be changed

config: {}
description: Default LXD profile
devices:
  eth0:
    name: eth0
    nictype: bridged
    parent: lxdbro
    type: nic
  root:
    path: /
    pool: zfs-pool
    size: 50GB
    type: disk
name: default
used_by: []
```


创建容器

镜像源（加速创建）清华的镜像源加速创建

```
sudo lxc remote add tuna-images  
https://mirrors.tuna.tsinghua.edu.cn/lxc-images/ --  
protocol=simplestreams --public
```

```
sudo lxc image list tuna-images:
```

创建ubuntu容器

```
sudo lxc launch tuna-images:ubuntu/18.04 test
```

进入容器

```
sudo lxc exec test bash
```

容器里的ubuntu已经存在了一个ubuntu用户

命令 `passwd ubuntu` 修改用户密码

容器里的ubuntu是一个很精简的系统，需要安装各种软件：

安装SSH: `apt install ssh`

因为我们没有设置桥接网卡，不能从外部电脑访问容器（不能ping通容器的ip），因此我们采用端口转发的方式来访问我们的容器，所以先退出容器`exit`

在宿主机
查看容器

```
icon@ubuntu:~$ sudo lxc list  
[sudo] password for icon:  
+-----+-----+-----+-----+  
| NAME   | STATE |         IPV4         |         |  
+-----+-----+-----+-----+  
| test   | RUNNING | 10.152.210.183 (eth0) | fd42:8997:bd92:c3 |  
+-----+-----+-----+-----+
```

通过 `ip addr` 查看宿主机ip地址：可知宿主机ip为
172.22.24.126

```
sudo iptables -t nat -A PREROUTING -d  
172.22.24.126 -p tcp --dport 60601 -j DNAT --to-  
destination 10.152.210.183:22
```

60610是我们定的端口号，通过宿主机的60601端口号映射到容器中22端口号（SSH默认端口号）

容器的配置

连接容器并配置：

```
ssh ubuntu@172.22.24.126 -p 60601
```

1. 更换网易的源

```
sudo mv /etc/apt/sources.list /etc/apt/sources.list.bak(备份)  
sudo vim /etc/apt/sources.list
```

将网易源添加进这个文件（注意系统版本 ubuntu 18.04）

```
deb http://mirrors.163.com/ubuntu/ bionic main restricted  
universe multiverse  
deb-src http://mirrors.163.com/ubuntu/ bionic main  
restricted universe multiverse
```

```
deb http://mirrors.163.com/ubuntu/ bionic-security main  
restricted universe multiverse  
deb-src http://mirrors.163.com/ubuntu/ bionic-security  
main restricted universe multiverse
```

```
deb http://mirrors.163.com/ubuntu/ bionic-updates main  
restricted universe multiverse  
deb-src http://mirrors.163.com/ubuntu/ bionic-updates  
main restricted universe multiverse
```

```
deb http://mirrors.163.com/ubuntu/ bionic-proposed main  
restricted universe multiverse  
deb-src http://mirrors.163.com/ubuntu/ bionic-proposed  
main restricted universe multiverse
```

```
deb http://mirrors.163.com/ubuntu/ bionic-backports main  
restricted universe multiverse  
deb-src http://mirrors.163.com/ubuntu/ bionic-backports  
main restricted universe multiverse
```

2 安装图形化界面

`sudo apt update`

`sudo apt install ubuntu-desktop gnome-panel gnome-settings-daemon metacity nautilus gnome-terminal -y`

3. 为容器添加显卡

我们回到宿主机

——为容器添加所有GPU: `lxc config device`

`add yourContainerName gpu gpu`

——添加指定GPU: `lxc config device add`

`yourContainerName gpu0 gpu id=0`

添加好显卡后，就相当于我们给容器安装了显卡

我们回到容器，然后安装显卡驱动

（与宿主机的显卡版本必须一致）

（虚拟机中没有英伟达的显卡，这里演示略过）

`sudo sh ./NVIDIA-Linux-X86_64-[YOURVERSION].run`

`--no-kernel-module`

容器里面安装显卡驱动时需要加上后面的参数，安装时不需要安装到内核

4. 远程连接

安装RDP脚本

`sudo apt install git`

（安装git后下载我们之后需要用的东西）

`git clone https://github.com/shenuiuin/LXD_GPU_SERVER`

打开文件夹

`cd LXD_GPU_SERVER/`

赋予脚本可执行权限

`sudo chmod +x Std-Xrdp-install-0.2.sh`

安装脚本

`./Std-Xrdp-install-0.2.sh`

```
#-----#
# Installation Completed
# Please test your xRDP configuration....
# Written by Griffon - April 2018 - Ver 0.2 - Std-Xrdp-Install-0.2.sh
#-----#
```

在安装好XRDP后，与之前一样，因为我们ping不通容器，所以我们需要将xrdp的端口转发到宿主机上

```
sudo iptables -t nat -A PREROUTING -d  
172.22.24.126 -p tcp --dport 60611 -j DNAT --to-  
destination 10.152.210.183:3389
```

60611是我们定的端口号，通过宿主机的60611端口号映射到容器中3389端口号（XRDP默认端口号）

可以通过windows的远程连接来使用容器(windows运行mstsc)



有一些图标的logo没有展示，没有关系给他换个主题就可以
接下来就是当普通的ubuntu来使用，
比如可以找一些教程：安装完ubuntu必做的事等

ubuntu美化

icon图标主题

```
sudo apt update
```

```
sudo apt install papirus-icon-theme
```

安装GTK主题

```
git clone https://github.com/vinceliuice/vimix-gtk-themes
```

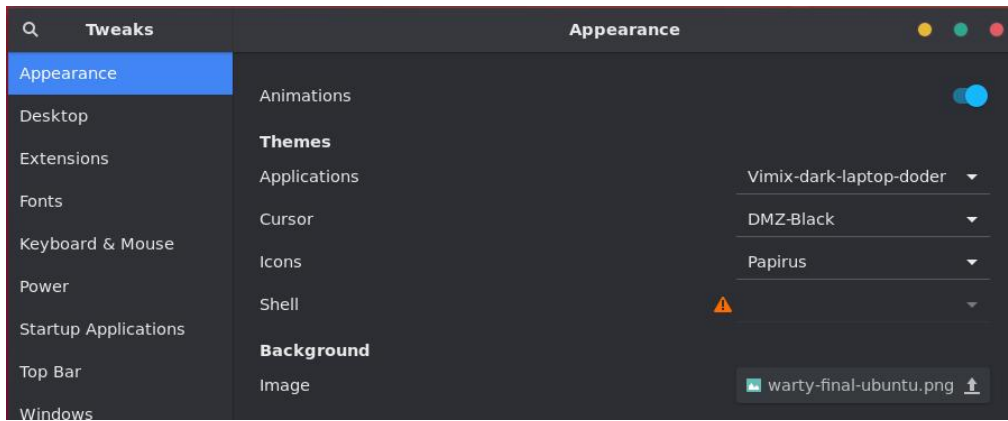
```
cd vimix-gtk-themes
```

```
sudo ./vimix-installer
```

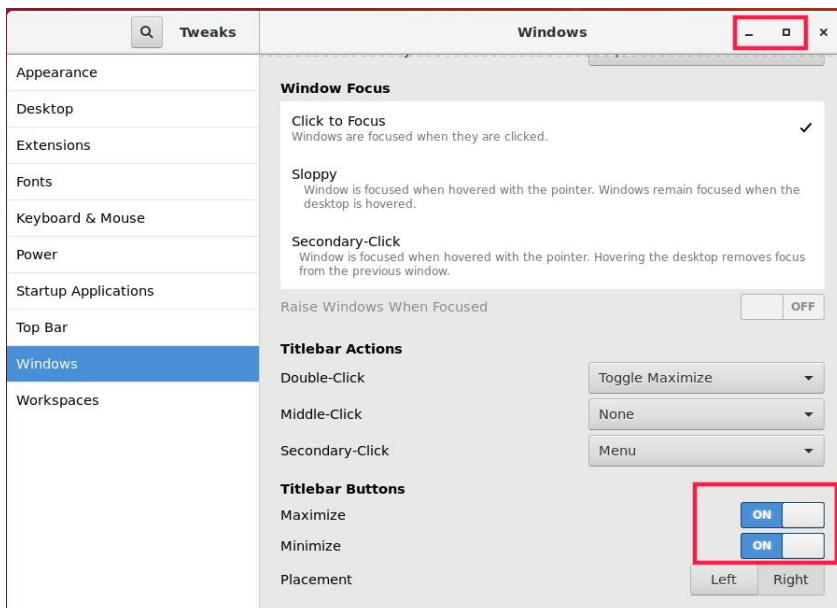
Tweaks安装:

```
sudo apt install gnome-tweak-tool -y
```

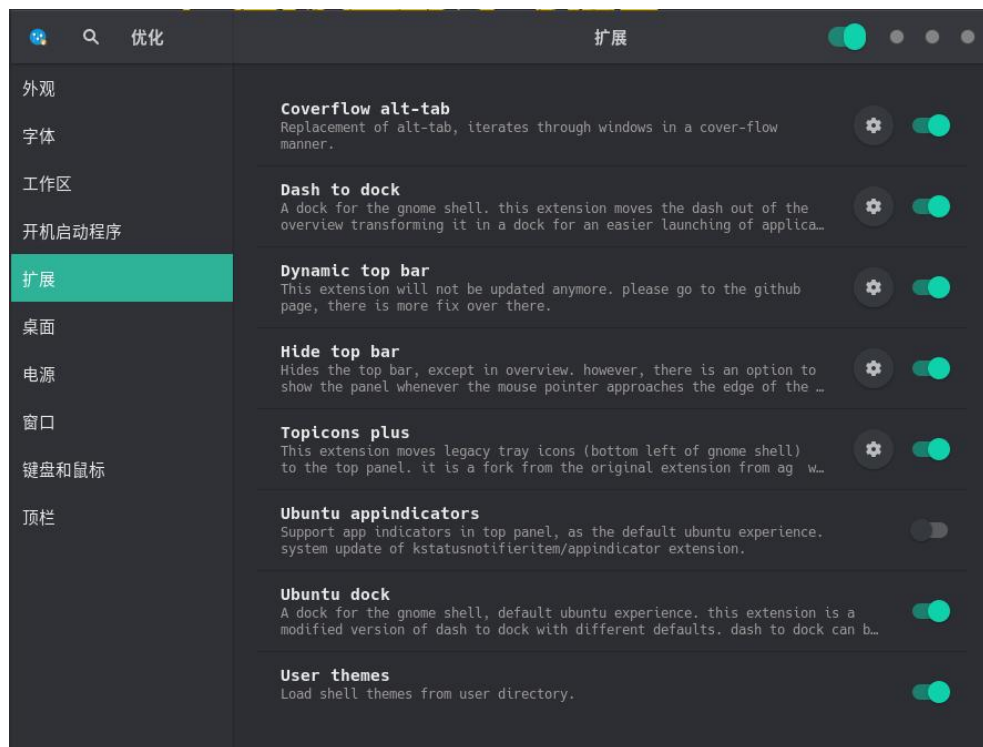
主题安装好后用Tweaks来应用主题



Tweaks可以个性化你的ubuntu，比如加上最大化最小化按钮



gnome扩展：
直接在ubuntu
的应用商店安装，
推荐的这些扩展



卸载自带的没用的软件

```
sudo apt-get remove --purge libreoffice*
```

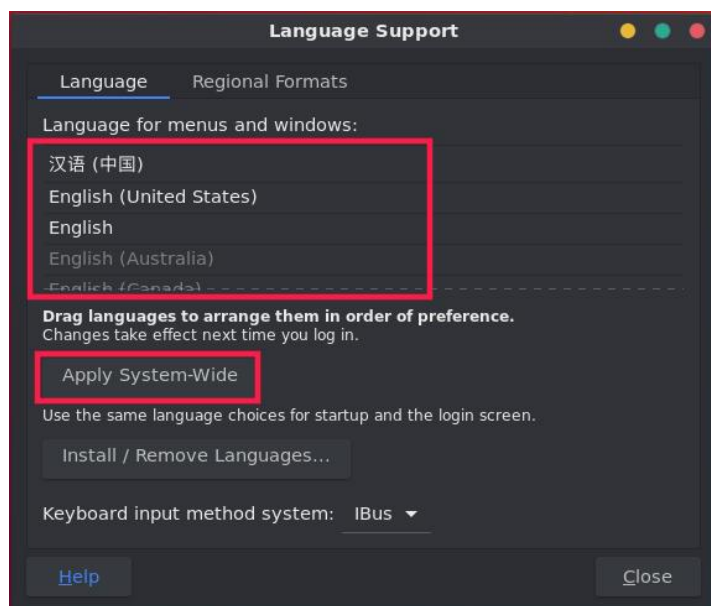
其他的可以在应用商店卸载

然后安装自己需要的软件

搜狗输入法、谷歌浏览器等等

之前已经安装好显卡驱动，现在只需要安装和宿
主机一样CUDA和cuDNN（版本必须相同）

系统的中文在Language Support。然后添加简体中文的语言
将中文拖到第一项，然后应用到整个系统。重启之后会有提示
将文件夹的名字改成中文（最好还是用旧的名称，英文路径）



容器管理

由于我们用端口转发的方式来连接容器，不过宿主机重启时会丢失路由表规则

#列出端口规则

```
sudo iptables -t nat -L PREROUTING --line-number
```

#删除第一行的规则

```
sudo iptables -t nat -D PREROUTING 1
```

保存规则（防止重启后转发表丢失）

```
sudo netfilter-persistent save
```

恢复保存的转发规则

```
sudo netfilter-persistent reload
```

lxdui



地址:

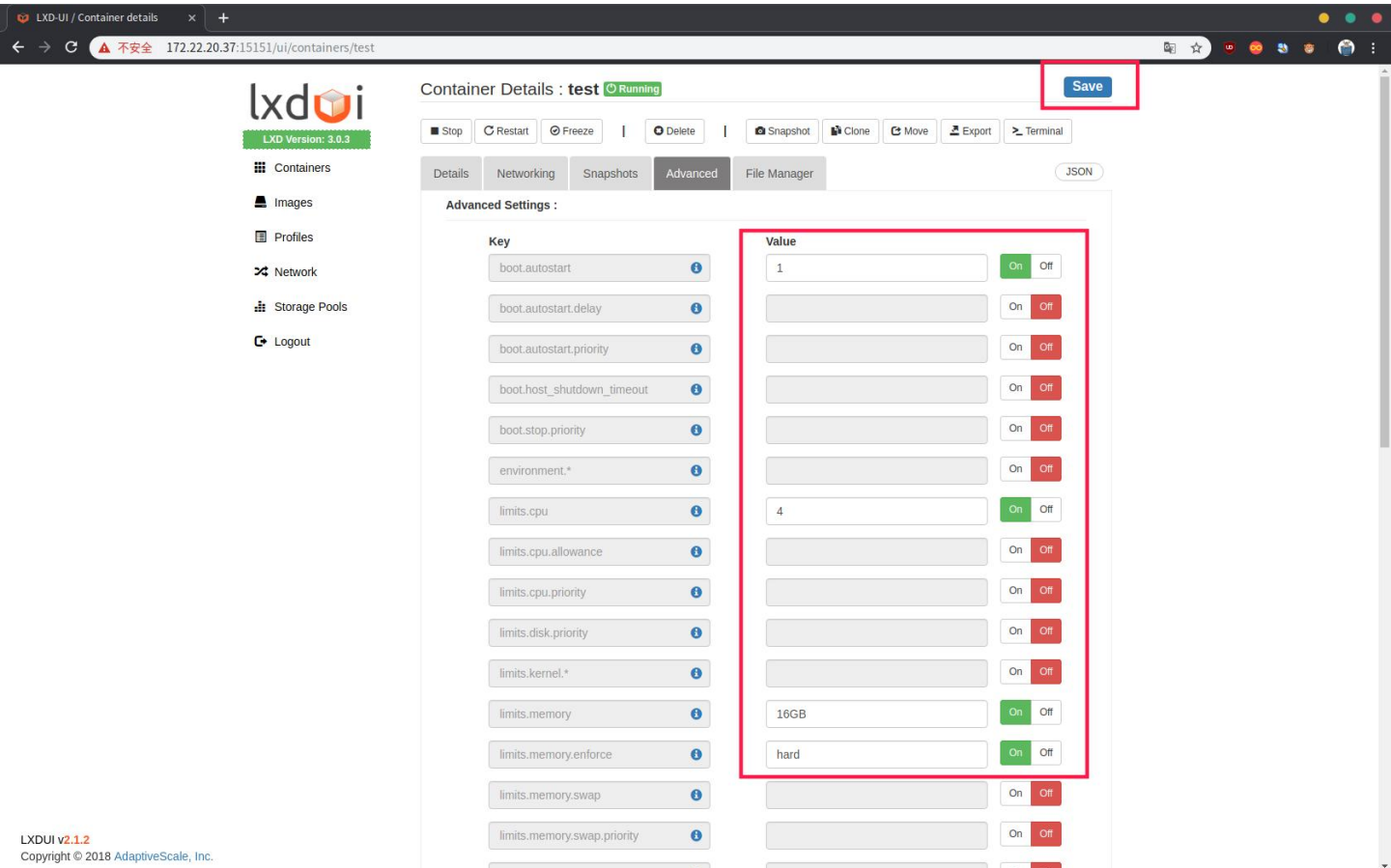
<https://github.com/AdaptiveScale/lxdui>

安装之前需要安装python3-dev，不然会出错

```
sudo apt install python3-dev
```

安装好后网页登录管理工具

http: (宿主机ip) :15151



在这个工具里面可以配置容器的各个参数，我们实验室的宿主机为256G内存，cpu48个核，容器主要用的是显卡，其他的参数按照人数平均分配一下，够用就可以

管理员应在桌面上新建一个文件，写下系统的版本、安装了什么软件、各种注意事项等等。

我们把这个配置好的容器当成模板，保存为镜像。

`sudo lxc stop test`

`sudo lxc publish test --alias ubuntu demo --public`

将test容器保存为ubuntu demo镜像 保存模板

以后直接用模板镜像来创建容器，容器创建好后还要为它添加显卡（驱动已经有了），还有用lxd ui配置它的参数，最后为它添加端口映射

```
icon@ubuntu:~$ sudo lxc publish test --alias ubuntu demo --public
Container published with fingerprint: b4bca2c4c4a6c11b7662d6397908e3131cd2d34d
icon@ubuntu:~$ sudo lxc image list
+-----+-----+-----+-----+
| ALIAS | FINGERPRINT | PUBLIC | DESCRIPTION |
+-----+-----+-----+-----+
| ubuntu demo | b4bca2c4c4a6 | yes | |
+-----+-----+-----+-----+
| | a6db29394c18 | no | Ubuntu bionic amd64 (20190412_08:15) |
+-----+-----+-----+-----+
icon@ubuntu:~$ sudo lxc launch b4bca2c4c4a6 name
```



```
student@mhms-deeplearning:~$ neofetch

.-/+oossssoo+/-.
`:+ssssssssssssssssss+:`
-+ssssssssssssssssssyyssss+-
.oSSsssssssssssssssdMMMNysssso.
/SSssssssssshdmmNNmyNMMMNhssssss/
+ssssssssshmydMMMMMMNdDDdyssssss+
/SSssssssshNMMMyhhyyyyhmNMMMNhssssss/
.sSSsssssdMMMNhssssssssshNMMMdssssss.
+SSssshhyNMMNysssssssssyNMMMyssssss+
ossyNMMMNyMMhssssssssssshmmhssssssso
ossyNMMMNyMMhssssssssssshmmhssssssso
+SSssshhyNMMNysssssssssyNMMMyssssss+
.sSSsssdMMMNhssssssssshNMMMdssssss.
/SSssssshNMMMyhhyyyyhdNMMMNhssssss/
+SSsssssdmydMMMMMMNdDDdyssssss+
/SSssssssshdmmNNmyNMMMNhssssss/
.oSSsssssssssssssssdMMMNysssso.
-+ssssssssssssssssssyyssss+-
`:+ssssssssssssssssss+:`
.-/+oossssoo+/-.

student@mhms-deeplearning
-----
OS: Ubuntu 18.04.2 LTS x86_64
Host: W780-G20 Grantley-EP
Kernel: 4.15.0-46-generic
Uptime: 2 days, 21 hours, 53 mins
Packages: 859
Shell: bash 4.4.19
Resolution: 3360x1050
WM: Muttter(DeepinGala)
Theme: Ambiance [GTK3]
Icons: Ubuntu-mono-dark [GTK3]
Terminal: /dev/pts/5
CPU: Intel Xeon E5-2650 v4 (48) @ 2.900GHz
GPU: NVIDIA TITAN V
GPU: ASPEED Technology, Inc. ASPEED Graphics Family
Memory: 42630MiB / 257843MiB
```

```
student@mhms-deeplearning:~$ lxc list
+-----+-----+-----+-----+
| NAME | STATE | IPV4 | IPV6 |
+-----+-----+-----+-----+
| lpl | RUNNING | 10.240.135.238 (eth0) | fd42:478d:c725:fca2:216:3eff:fef3:914f (eth0) | PE
+-----+-----+-----+-----+
| sh | RUNNING | 10.240.135.179 (eth0) | fd42:478d:c725:fca2:216:3eff:fe16:7d54 (eth0) | PE
+-----+-----+-----+-----+
| swz | RUNNING | 10.240.135.168 (eth0) | fd42:478d:c725:fca2:216:3eff:fe4d:50bd (eth0) | PE
+-----+-----+-----+-----+
| test | STOPPED | | | PE
+-----+-----+-----+-----+
| ubuntu | STOPPED | | | PE
+-----+-----+-----+-----+
| wnbot | STOPPED | | | PE
+-----+-----+-----+-----+
| yql | RUNNING | 10.240.135.170 (eth0) | fd42:478d:c725:fca2:216:3eff:fe2c:1c29 (eth0) | PE
+-----+-----+-----+-----+
| yx | RUNNING | 10.240.135.139 (eth0) | fd42:478d:c725:fca2:216:3eff:fe64:3944 (eth0) | PE
+-----+-----+-----+-----+
| zsy | RUNNING | 10.240.135.54 (eth0) | | PE
+-----+-----+-----+-----+
| ztt | RUNNING | 10.240.135.195 (eth0) | fd42:478d:c725:fca2:216:3eff:fe5f:5ecd (eth0) | PE
+-----+-----+-----+-----+
student@mhms-deeplearning:~$
```

活动

Saturday 06 : 46

read.txt

回收站

ubuntu@sh: ~

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

→ ~ nvidia-smi

Sat Apr 13 06:45:38 2019

NVIDIA-SMI 410.78		Driver Version: 410.78		CUDA Version: 10.0	
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util Compute M.
0	TITAN V	Off	00000000:04:00.0	Off	N/A
52%	71C	P2	182W / 250W	4755MiB / 12036MiB	100% Default
1	TITAN V	Off	00000000:08:00.0	Off	N/A
58%	79C	P2	219W / 250W	3720MiB / 12036MiB	100% Default
2	TITAN V	Off	00000000:09:00.0	Off	N/A
66%	84C	P2	158W / 250W	3720MiB / 12036MiB	100% Default
3	TITAN V	Off	00000000:82:00.0	Off	N/A
62%	83C	P2	180W / 250W	3720MiB / 12036MiB	100% Default
4	TITAN V	Off	00000000:85:00.0	Off	N/A
62%	83C	P2	178W / 250W	3720MiB / 12036MiB	100% Default
5	TITAN V	Off	00000000:86:00.0	Off	N/A
62%	83C	P2	190W / 250W	3720MiB / 12036MiB	100% Default
6	TITAN V	Off	00000000:89:00.0	Off	N/A
57%	77C	P2	184W / 250W	3720MiB / 12036MiB	100% Default
7	TITAN V	Off	00000000:8A:00.0	Off	N/A
62%	83C	P2	157W / 250W	3720MiB / 12036MiB	100% Default

Processes:

GPU	PID	Type	Process name
-----	-----	------	--------------

ubuntu@sh: ~

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

ubuntu@sh % neofetch

student@mhms-deeplearning

OS: Ubuntu 18.04.2 LTS x86_64

Host: W780-G20 Grantley-EP

Kernel: 4.15.0-46-generic

Uptime: 3 days, 3 hours, 32 mins

Packages: 1880

Shell: zsh 5.4.2

Resolution: 1680x1050

DE: GNOME 3.28.3

WM: GNOME Shell

WM Theme: Adwaita

Theme: Vimbix-dark-beryl [GTK2/3]

Icons: Oranchole-Classic-Folders [GT

Terminal: gnome-terminal

CPU: Intel Xeon E5-2650 v4 (4) @ 2.9

GPU: ASPEED Technology, Inc. ASPEED

GPU: NVIDIA TITAN V

Memory: 2511MiB / 15258MiB