**配置ubuntu版的lxc容器**

lxc是一种系统层级的虚拟化方案，利用clone这个系统调用的一些特性，模拟一个虚拟化的场景（独立的网络、文件系统），并利用cgroup进行资源的限制管理。

和常见的xen、kvm不同，它不是一种半虚拟化技术，他仅仅使用了本地cpu，只有一个传统意义上的操作系统。类似的解决方案有Solaris的Zone，在Freebsd的Jail。

网络上关于ubuntu的lxc容器都有些小问题，所以整理一下希望对大家有些帮助，lxc的相关命令就不解释了，比较简单。

#### kernel

基本上kernel >= 2.6.27的linux下都已经在内核态支持lxc。

使用lxc-checkconfig进行检测

hoterran@hoterran-laptop:~$ sudo lxc-checkconfig

Kernel config /proc/config.gz not found, looking in other places...

Found kernel config file /boot/config-2.6.35-28-generic

--- Namespaces ---

Namespaces: enabled

Utsname namespace: enabled

Ipc namespace: enabled

Pid namespace: enabled

User namespace: enabled

Network namespace: enabled

Multiple /dev/pts instances: enabled

--- Control groups ---

Cgroup: enabled

Cgroup namespace: enabled

Cgroup device: enabled

Cgroup sched: enabled

Cgroup cpu account: enabled

Cgroup memory controller: enabled

Cgroup cpuset: enabled

--- Misc ---

Veth pair device: enabled

Macvlan: enabled

Vlan: enabled

确认每个选项都是enabled状态即可。

#### lxc tools

除了内核支持外还需要一些用户态的工具。

sudo apt-get install lxc bridge-utils debootstrap libcap-dev

另外bridge-utils搭建网桥的工具，debootstrap是用来搭建一个基本的debain容器系统的工具。

通过一下命令判断lxc是否安装正常。

hoterran@hoterran-laptop:~$ sudo lxc-execute -n test /bin/echo "hello world"

hello world

如果你迫不及待想看看lxc的效果那么你可以试试如下的命令

/usr/bin/lxc-execute -n foo -f /usr/share/doc/lxc/examples/lxc-macvlan.conf /bin/bash

#### cgroups

cgroup是资源管理的关键。

mkdir /cgroup

mount none -t cgroup /cgroup

#### bridges

网桥打通主机和容器的网络。  
环境如下  
192.169.1.\* 是你的主机的地址，我们不去管他。  
192.168.2.1 br网桥的地址，192.168.2.2 lxc容器的地址。  
配置网桥，这里假定eth0作为连通的物理网卡，这里的eth0和后面的config.ubuntu文件要一一对应。

vim /etc/network/interfaces

auto br0

iface br0 inet static

address 192.168.2.1

netmask 255.255.255.0

broadcast 192.168.2.255

gateway 192.168.2.1

bridge\_ports eth0

bridge\_stp off

bridge\_maxwait 5

post-up /usr/sbin/brctl setfd br0 0

是链路层面的交换，所有把ip给去掉。

ifconfig eth0 0.0.0.0

重启网络激活网桥

service networking stop

service networking start

观察网桥的状况

hoterran@hoterran-laptop:~$ sudo brctl show

bridge name bridge id STP enabled interfaces

br0 8000.f0def12e0cdf no eth0

也可以通过brctl直接添加网桥，这个更简单。

#### lxc 容器的配置

创建容器的目录

mkdir /lxc

cd lxc

先创建一个mini版的ubuntu系统

sudo debootstrap --variant=minbase --arch i386 lucid rootfs.ubuntu http://ubuntu.srt.cn/ubuntu

...

...

...

这里要等待一会，要下载大量的文件。

容器是简历在系统之上的，无法和驱动打交道没法认识到任何硬件，所以我们自己产生dev目录下的设备文件。  
我们来编写一个脚本自动产生这些设备文件。

vim /usr/local/bin/lxc-config

ROOT=$(pwd)

DEV=${ROOT}/dev

if [ $ROOT = '/' ]; then

printf "033[22;35mnDO NOT RUN ON THE HOST NODEnn"

tput sgr0

exit 1

fi

if [ ! -d $DEV ]; then

printf "033[01;33mnRun this script in rootfsnn"

tput sgr0

exit 1

fi

rm -rf ${DEV}

mkdir ${DEV}

mknod -m 666 ${DEV}/null c 1 3

mknod -m 666 ${DEV}/zero c 1 5

mknod -m 666 ${DEV}/random c 1 8

mknod -m 666 ${DEV}/urandom c 1 9

mkdir -m 755 ${DEV}/pts

mkdir -m 1777 ${DEV}/shm

mknod -m 666 ${DEV}/tty c 5 0

mknod -m 666 ${DEV}/tty0 c 4 0

mknod -m 666 ${DEV}/tty1 c 4 1

mknod -m 666 ${DEV}/tty2 c 4 2

mknod -m 666 ${DEV}/tty3 c 4 3

mknod -m 666 ${DEV}/tty4 c 4 4

mknod -m 600 ${DEV}/console c 5 1

mknod -m 666 ${DEV}/full c 1 7

mknod -m 600 ${DEV}/initctl p

mknod -m 666 ${DEV}/ptmx c 5 2

exit 0

再进入系统的目录运行这个脚本

chmod u+x /usr/local/bin/lxc-config

cd /lxc/root.ubuntu

sudo sh /usr/local/bin/lxc-config

设备产生完毕，我们来配置容器，我们使用chroot跳到这个容器所在目录的内部，避免改了宿主机的文件

chroot /lxc/rootfs.ubuntu /bin/bash

下载有些必须的程序，尤其sshd，我们稍后通过它来连接到lxc容器。

apt-get install openssh-server vim rsyslog sudo

设置密码

passwd

修改一些基本的配置文件

rm /etc/mtab

ln -s /proc/mounts /etc/mtab

echo "hoterran\_lxc" > /etc/hostname

echo "127.0.0.1 localhost" > /etc/hosts

下面的dns一定要设置正确，否则ssh会变慢

vim /etc/resolve.conf

nameserver 192.168.2.1

修改init文件，这里主意，启动lxc容器和启动物理机有很大不同。我们只需要配置/etc/init/rc-sysinit.conf和/etc/init.d/rc.lxc文件。

/etc/init只保留如下3个文件。

hoterran@hoterran-laptop:/lxc/rootfs.ubuntu2/etc/init$ ls -al

total 20

drwxr-xr-x 2 root root 4096 2011-10-03 15:46 .

drwxr-xr-x 48 root root 4096 2011-10-03 15:38 ..

-rwxr-xr-x 1 root root 559 2011-10-03 15:46 rc-sysinit.conf

-rw-r--r-- 1 root root 298 2011-10-03 15:40 rsyslog.conf

-rw-r--r-- 1 root root 632 2011-10-03 15:40 ssh.conf

ssh.conf，rsyslog.conf是自动产生的，rc-sysinit.conf是容器的启动文件，需要我们来编辑。

vim /etc/init/rc-sysinit.conf

# rc - System V runlevel compatibility

#

# This task runs the old System V-style rc script when changing between

# runlevels.

start on startup

task

pre-start script

mkdir -p /var/run/network

touch /var/run/utmp

chmod 664 /var/run/utmp

chown root.utmp /var/run/utmp

chmod -x /etc/network/\*/upstart 2>/dev/null

route add default gw 192.168.2.1

end script

script

start networking

initctl emit filesystem --no-wait

initctl emit local-filesystems --no-wait

initctl emit virtual-filesystems --no-wait

init 2

end script

exec /etc/init.d/rc.lxc

/etc/init.d/rc.lxc为启动服务的文件

vim /etc/init.d/rc.lxc

#!/bin/bash

/etc/init.d/rsyslog start &

/etc/init.d/ssh start &

退出chroot

exit

这样整个容器内部的操作系统就配置完毕了,回到宿主机，还是在/lxc目录，我们来配置lxc容器的启动配置文件  
vim config.ubuntu

lxc.utsname = ubuntu

lxc.tty = 4

lxc.network.type = veth

lxc.network.flags = up

lxc.network.link = br0

lxc.network.name = eth0

lxc.network.mtu = 1500

lxc.network.ipv4 = 192.168.2.2/24

lxc.rootfs = /lxc/rootfs.ubuntu

lxc.mount = /lxc/fstab.ubuntu

lxc.cgroup.devices.deny = a

# /dev/null and zero

lxc.cgroup.devices.allow = c 1:3 rwm

lxc.cgroup.devices.allow = c 1:5 rwm

# consoles

lxc.cgroup.devices.allow = c 5:1 rwm

lxc.cgroup.devices.allow = c 5:0 rwm

lxc.cgroup.devices.allow = c 4:0 rwm

lxc.cgroup.devices.allow = c 4:1 rwm

# /dev/{,u}random

lxc.cgroup.devices.allow = c 1:9 rwm

lxc.cgroup.devices.allow = c 1:8 rwm

# /dev/pts/\* - pts namespaces are "coming soon"

lxc.cgroup.devices.allow = c 136:\* rwm

lxc.cgroup.devices.allow = c 5:2 rwm

# rtc

lxc.cgroup.devices.allow = c 254:0 rwm

其中网络相关的文件要尤其注意不要配置错误。

设置容器启动后需要挂载的文件系统  
vim fstab.ubuntu

none /lxc/rootfs.ubuntu/proc proc defaults 0 0

none /lxc/rootfs.ubuntu/sys sysfs defaults 0 0

none /lxc/rootfs.ubuntu/var/lock tmpfs defaults 0 0

none /lxc/rootfs.ubuntu/var/run tmpfs defaults 0 0

none /lxc/rootfs.ubuntu/dev/pts devpts defaults 0 0

到这里为止宿主机环境，我们创建了一个容器目录，配置了两个文件。

hoterran@hoterran-laptop:/lxc$ ls -al

total 52

drwxr-xr-x 4 root root 4096 2011-10-03 15:48 .

drwxr-xr-x 24 root root 4096 2011-10-01 15:55 ..

-rw-r--r-- 1 root root 791 2011-10-01 21:37 1

-rw-r--r-- 1 root root 789 2011-10-03 15:48 config.ubuntu

-rw-r--r-- 1 root root 252 2011-10-01 21:59 fstab.ubuntu

drwxr-xr-x 20 root root 4096 2011-10-03 11:49 rootfs.ubuntu

#### 启动容器

lxc-create -n ubuntu -f config.ubuntu

lxc-start -n ubuntu

#### 登录到容器

ssh root@192.168.2.2