Raspberry Pi - Support #576

Creating Linux Containers on Arch Linux for Raspberry Pi

02/27/2015 11:20 PM - Daniel Curtis

Status:	Closed	**Start date:**	02/27/2015
Priority:	Normal	**Due date:**	
Assignee:	Daniel Curtis	**% Done:**	100%
Category:	Jails / Containers	**Estimated time:**	3.00 hours
Target version:	Arch Linux	**Spent time:**	6.50 hours

Description

This is a simple guide for setting up and using Linux Containers on Arch Linux.

Prepare the Environment

- Make sure the system is up to date:

```
pacman -Syu
```

- Install the base-devel and wget packages:

```
pacman -S base-devel wget rsync
```

- Install yaourt

Set Up The Network

- Install netctl:

```
pacman -S netctl
```

Bridged Wired Connection

- Bridge Internet-shared - This example will bridge network interface eth0 and configure a static IP for the bridge:

```
nano /etc/netctl/lxcbridge
```

  - And add/modify the following:

```
Description="LXC Bridge"
Interface=br0
Connection=bridge
BindsToInterfaces=(eth0)
IP=static
Address=192.168.1.100/24
Gateway=192.168.1.1
DNS=192.168.1.1
SkipForwardingDelay=yes
```

- After changes are made, make sure to re-enable and restart the bridge:

```
netctl enable lxcbridge
```

```
netctl start lxcbridge
```

- Enable IP Forwarding persist at boot:

```
echo 'net.ipv4.ip_forward=1' >> /etc/sysctl.d/40-ip-forward.conf
```

- And also apply this iptables rule:

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

    - To make changes persist upon reboot:

```
iptables-save > /etc/iptables/iptables.rules
systemctl enable iptables
systemctl start iptables
```

NAT Wireless Connection

- Make sure that forwarding is turned on to support NAT:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

- Make the previous change persistent after a reboot:

```
echo 'net.ipv4.ip_forward=1' >> /etc/sysctl.d/30-ipforward.conf
echo 'net.ipv6.conf.default.forwarding=1' >> /etc/sysctl.d/30-ipforward.conf
echo 'net.ipv6.conf.all.forwarding=1' >> /etc/sysctl.d/30-ipforward.conf
```

Now create the bridge:

- NAT Internet-shared - This example will bridge network interface eth0 and configure a static IP for the bridge:

```
nano /etc/netctl/lxcnatbridge
```

    - And add/modify the following:

```
Description="LXC NAT Bridge"
Interface=natbr0
Connection=bridge
IP=static
Address=192.168.10.200/24
DNS=192.168.1.1
SkipForwardingDelay=yes
ExecUpPost="/usr/local/bin/natbr0-up"
```

Create the nat script:

```
nano /usr/local/bin/natbr0-up
```

- And add the following:

```
#!/bin/sh
```

```
# Script to setup NAT iptables masquerade rules; and dnsmasq for DHCP and DNS.

# This is the address assigned to the bridge at boot
BRADDR=192.168.10.200

# DHCP IP address range for containers
BRRANGE=192.168.10.201,192.168.10.250

# Configure iptables rules for NAT
iptables -A FORWARD -i natbr0 -s ${BRADDR}/24 -m conntrack --ctstate NEW -j ACCEPT
iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
iptables -A POSTROUTING -t nat -j MASQUERADE

# Start dnsmasq DNS/DHCP server for the network attached to the NAT interface.
dnsmasq --bind-interfaces --conf-file= --listen-address $BRADDR --except-interface lo --dhcp-r
ange $BRRANGE --dhcp-lease-max=253 --dhcp-no-override
```

- Make sure it is executable by doing

```
chmod +x /usr/local/bin/natbr0-up
```

- Start and enable the NAT bridge at boot:

```
netctl enable lxcnatbridge
netctl start lxcnatbridge
```

- Now a container lxc.conf should read:

```
lxc.network.type=veth
lxc.network.link=natbr0
lxc.network.flags=up
```

Install LXC

- Install lxc, bridge-utils, and arch-install-scripts:

```
pacman -S bridge-utils arch-install-scripts dnsmasq
```

  - (Optional) Install extra packages for lxc:

```
pacman -S lua lua-filesystem lua-alt-getopt
```

- Install debootstrap from AUR:

```
yaourt debootstrap
```

  - Install gnupg1 for keyring verification; make sure to edit the PKGBUILD and add armv7h to the arch parameter:

```
yaourt gnupg1
```

  - Install debian-archive-keyring:

```
yaourt debian-archive-keyring
```

- Install ubuntu-keyring:

```
yaourt ubuntu-keyring
```

- Test that the system is correctly configured

```
lxc-checkconfig
```

- The output should be similar to:

```
--- Namespaces ---
Namespaces: enabled
Utsname namespace: enabled
Ipc namespace: enabled
Pid namespace: enabled
User namespace: missing
Network namespace: enabled
Multiple /dev/pts instances: enabled

--- Control groups ---
Cgroup: enabled
Cgroup clone_children flag: 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
File capabilities: enabled
```

Container setup

To find all available templates that come with LXC, look in /usr/share/lxc/templates directory:

```
ls /usr/share/lxc/templates
```

- *Example output:*

```
lxc-alpine  lxc-altlinux  lxc-archlinux  lxc-busybox  lxc-centos  lxc-cirros  lxc-debian  lxc-
download  lxc-fedora  lxc-gentoo  lxc-openmandriva  lxc-opensuse  lxc-oracle  lxc-plamo  lxc-s
shd  lxc-ubuntu  lxc-ubuntu-cloud
```

Arch Container

- Create the container:

```
lxc-create -n arch.example.com -t archlinux
```

- Edit the Arch Linux container config file:

```
nano /var/lib/lxc/arch.example.com/config
```

- And add/modify the following:

```
lxc.network.type = veth
lxc.network.flags = up
lxc.network.link = br0
#lxc.network.hwaddr =
lxc.network.ipv4 = 192.168.1.101
lxc.network.ipv4.gateway = 192.168.1.1
lxc.network.name = eth0
lxc.autodev = 1
lxc.pts = 1024
lxc.kmsg = 0
lxc.rootfs = /var/lib/lxc/arch.example.com/rootfs
lxc.utsname = arch.example.com
lxc.arch = armv7l
```

- Start the Arch container:

```
lxc-start -n arch.example.com
```

- Open a console:

```
lxc-attach -n arch.example.com
```

- And change the password:

```
passwd
```

- Create a wired connection:

```
cp /etc/netctl/examples/ethernet-static /etc/netctl/wired
```

- Edit the /etc/netctl/wired to match your needs.

```
nano /etc/netctl/wired
```

- Add/modify the following:

```
Description='Ethernet Connection'
Interface=eth0
Connection=ethernet
IP=static
Address=('192.168.1.101/24')
Gateway=('192.168.1.1')
DNS=('192.168.1.1')

# Required to start a network connection in a container
ForceConnection=yes
```

- Start and enable the wired connection at boot:

```
netctl enable wired
```

- **NOTE**: I needed to edit /etc/systemd/system/netctl\@wired.service and comment out the following in order for the netctl command to start:

```
#BindsTo=sys-subsystem-net-devices-eth0.device
#After=sys-subsystem-net-devices-eth0.device
```

  - Then reload the systemd units:

```
systemctl daemon-reload
```

- While the console to the container is open, install openssh

```
pacman -S openssh
```

  - Start and enable openssh at boot:

```
systemctl enable sshd.service
systemctl start sshd.service
```

Raspbian Container

- Copy the existing Debian LXC template:

```
cp /usr/share/lxc/templates/lxc-debian /usr/share/lxc/templates/lxc-raspbian
```

- Edit the Raspbian LXC template:

```
nano /usr/share/lxc/templates/lxc-raspbian
```

  - And modify the following parameters:

```
MIRROR=${MIRROR:-http://archive.raspbian.org/raspbian}
#...
arch='armhf'
debootstrap --verbose --variant=minbase --arch=$arch --no-check-gpg
```

  - **NOTE**: The MIRROR variable is set to Rasbian repositories at http://archive.raspbian.org/raspbian
  - **NOTE**: The arch variable is set to **armhf**
  - **NOTE**: The debootstrap command has the added **--no-check-gpg argument**

- Create the container:

```
lxc-create -n raspbian.example.com -t raspbian
```

- Edit the Raspbian container config file:

```
nano /var/lib/lxc/raspbian.example.com/config
```

  - And add/modify the following:

```
lxc.network.type = veth
lxc.network.flags = up
lxc.network.link = natbr0
```

```
lxc.network.name = eth0
lxc.rootfs = /var/lib/lxc/raspbian.example.com/rootfs
lxc.utsname = raspbian.example.com
lxc.arch = armvhf
```

- Start the containter:

```
lxc-start -n raspbian.example.com
```

- Configure the container to use systemd within lxc:

```
lxc.autodev = 1
lxc.pts = 1024
lxc.kmsg = 0
lxc.hook.autodev=/var/lib/lxc/raspbian.example.com/autodev
```

Autostart Containers

- Autostart the Arch container:

```
systemctl enable lxc@arch.example.com.service
```

- Autostart the Raspbian container:

```
systemctl enable lxc@raspbian.example.com.service
```

Resources

- https://s3hh.wordpress.com/2011/05/17/lxc-containers-on-a-host-with-wireless/
- https://wiki.archlinux.org/index.php?title=Linux_Containers&redirect=no

History

#1 - 03/01/2015 01:03 PM - Daniel Curtis

- % Done changed from 10 to 30

- Description updated

- Status changed from New to In Progress

#2 - 03/01/2015 05:34 PM - Daniel Curtis

- Description updated

#3 - 03/01/2015 06:34 PM - Daniel Curtis

- Description updated

#4 - 03/01/2015 06:39 PM - Daniel Curtis

- Description updated

- % Done changed from 30 to 50

#5 - 03/01/2015 07:20 PM - Daniel Curtis

- Description updated

#6 - 03/02/2015 02:57 PM - Daniel Curtis

- Description updated

#7 - 04/14/2015 02:53 PM - Daniel Curtis

- Description updated

#8 - 05/05/2015 12:49 PM - Daniel Curtis

- Description updated

- % Done changed from 50 to 70

#9 - 05/05/2015 01:01 PM - Daniel Curtis

- Description updated

#10 - 05/05/2015 01:13 PM - Daniel Curtis

- Description updated

#11 - 05/05/2015 04:38 PM - Daniel Curtis

- Description updated

#12 - 11/15/2015 02:46 PM - Daniel Curtis

- Description updated

- Status changed from In Progress to Resolved

- % Done changed from 70 to 100

#13 - 11/27/2015 04:47 PM - Daniel Curtis

- Status changed from Resolved to Closed

#14 - 07/15/2016 07:42 PM - Daniel Curtis

- Description updated