Building Docker from source on the Raspberry Pi

I've been exploring the world of linux containers a bit lately and so I thought I'd write a quick tutorial on how to set up the Raspberry Pi (my go to remote embedded build machine) to build Docker from source. If you don't know what Docker is then you should do yourself a favour and read up about it from their site. It's a truly amazing piece of software and I'm hoping to use it for instances of node applications as well as embedded build instances in the future. For now lets just build it. As a forewarning I have built my own 3.12 kernel for this particular Raspberry Pi so if something doesn't work on the default Raspbian image that could be why.

First up install the userspace tools for linux containers. You could of course install the 'lxc' package from the repositories but at this time it is on version 0.8.0-rc1 and as I'm building everything bleeding edge I'm sure you'll need the latest version.

```
sudo apt-get install libcap-dev
cd /opt
sudo git clone https://github.com/lxc/lxc.git
cd lxc
sudo ./autogen.sh && sudo ./configure && sudo make && sudo make install
lxc-version
lxc version: 1.0.0.beta1
```

Next we need to get go. Again we could install the 'golang' package but this is the 1.0.2 version and 1.1 version contains many improvements for ARM. So I've chosen to get a precompiled version.

```
cd /opt
sudo wget -c http://dave.cheney.net/paste/go1.2rc1.linux-arm~multiarch-armv6-1.tar.gz
sudo tar -C /usr/local -xzf go1.2rc1.linux-arm~multiarch-armv6-1.tar.gz
Add export PATH=$PATH:/usr/local/go/bin to your ~/.profile or /etc/profile
go version
go version devel +5b367120d592 Thu Sep 19 17:12:00 2013 +1000 linux/arm
```

Building from source is also pretty easy

```
curl https://go.googlecode.com/files/go1.2.src.tar.gz | tar xz -C /usr/local
cd /usr/local/go/src
```

```
./make.bash
go version go1.2 linux/arm
```

I have included the AUFS patches from here into my kernel but from docker version 0.7 you do not explicitly need them. Patching is done by compying in the new sources and then running the patches from the root of your kernel source tree.

AUFS-Utils is also useful if you plan on using this feature.

```
cd /opt
sudo git clone git://git.code.sf.net/p/aufs/aufs-util
cd aufs-util
sudo git checkout aufs3.x-rcN
sudo make && sudo make install
```

The most difficult part of this whole process is the fact that the docker build system uses docker containers to build itself so you need to have docker to build docker. The guys at resin.io have made things a whole lot easier by publishing a raspberry pi built docker binary. This can be found here, extracted and used.

```
docker -v
Docker version 0.7.2, build 28b162e-dirty
```

Building from source can then be completed using the standard installation instructions from the docker website.