New version!

It's been a long time coming, but there's now a neat copy of my Centos7 container
https://github.com/DruidFish/docker-cvmfs-athena☐

Old version below

So, you thought you should try out Docker

Yep, me too. It's buzz-wordy software and you don't want to be an old stick-in-the mud, right? It's a way to get a particular software environment set up quickly, and it's not clunky like a Virtual Machine, so you reckoned you could use it to run software on your own machine rather than dealing with LXPLUS or whatever.

Good news! It works! And I wrote a quick guide on how to use it, so you don't have to work this out for yourself.

Just tell me what to do already

OK, here's the zero-explanation recipe. First you'll have to install Docker. For me (using Ubuntu) this was just

```
sudo apt install docker.io
```

But obviously this part will vary depending on your Operating System. The official guide is here☐

Now you need to start a Docker container. We want the CERN SLC6 environment, like you get on LXPLUS:

```
docker pull cern/slc6-base
mkdir MyDockerTest
cd MyDockerTest
mkdir workdir
docker run --privileged -i -t -v $PWD/workdir:/workdir cern/slc6-base /bin/bash
```

You might need to run the Docker commands as root (i.e. sudo docker whatever).

Good news: now you're in a Docker container, as the root user, and it's pretending to be SLC6. Bad news: most of the software you want isn't there. Docker is lightweight because it comes with the bare minimum of things. First we need CVMFS, the networked filesystem that provides most of our software. The following assumes you work on ATLAS:

```
yum install https://ecsft.cern.ch/dist/cvmfs/cvmfs-release/cvmfs-release-latest.noarch.rpm
yum install cvmfs cvmfs-config-default
yum install gcc.x86_64
echo 'CVMFS_REPOSITORIES=atlas.cern.ch,atlas-condb.cern.ch,grid.cern.ch,atlas-nightlies.cern.ch,s
echo 'CVMFS_HTTP_PROXY="DIRECT"' >> /etc/cvmfs/default.local
cvmfs_config setup
cvmfs_config probe
```

Assuming you got an "OK" message at the end of each of the CVMFS tests you're good to go! Set up your software environment as if you were on LXPLUS:

```
export ATLAS_LOCAL_ROOT_BASE=/cvmfs/atlas.cern.ch/repo/ATLASLocalRootBase
source $ATLAS_LOCAL_ROOT_BASE/user/atlasLocalSetup.sh
```

For LHCb you'll need to do this last part differently, obviously, as well as putting some different CVMFS repositories into your setup earlier. Also note that ATLAS software seems to need a local version of GCC as

well as the one from CVMFS, so I installed that in the earlier section too.

How do I keep my shiny new container?

Docker has "images" that define an operating system or software environment, and "containers" that are one running copy of a particular image. While your container is running, open a new terminal on your host machine and try this:

```
docker container ls
```

Again, you might need to try "sudo docker" instead.

Which will hopefully give you a result like this

```
CONTAINER ID        IMAGE              COMMAND           CREATED             STATUS
bb169f195cdc        cern/slc6-base     "/bin/bash"       About an hour ago   Up 9 minutes
```

The name at the end is the interesting part: it's a random name for the container you just made. Make a note of it! Mine is "boring_hopper" but yours will be something else.

You can now quit your running container with "exit" as though it was an SSH session, and restart it with

```
docker start -ai boring_hopper
```

With your own container name instead of "boring_hopper".

I re-run the cvmfs_config commands (setup and probe) from earlier each time I restart my container, but I get the feeling it's not necessary.

Is that everything?

Hell no. There's a huge amount of functionality here that I haven't even looked at. A big part of Docker is the website DockerHub⧉ which is where you got your SLC6 environment from earlier. You can download different images, upload your own creations, and there's lots of other stuff too. Have fun!

One thing you might have noticed earlier is that I made a directory called "workdir" that exists on your host machine, but is mounted in your Docker container as "/workdir". I like to do all my work in this folder so I can access any code bits I want from the host machine later.

Missing Docker software

Try this if needed

```
yum install zlib zlib-devel uuid uuid-devel freetype
```

I actually had to copy libuuid and uuid.h from LXPLUS, since the versions I got from yum weren't compatible with Gaudi.

Recently when building the whole of Athena, I also found that I needed

```
yum install mesa-libGL-devel libxml2-devel patch
```

-- BenjaminWynne - 2017-11-01

This topic: Main > AtlasEdinburghDockerGuide
Topic revision: r6 - 2021-04-21 - BenjaminWynne