Advanced Docker Course

Dockerizing an application

Different kinds of application:

- command line program, works on files (samtools)
- long-running server (mysql), communicates via network
- virtual machine (ubuntu)

We will focus on the first type

Installing programs

Building an image is like administering a Linux computer

```
apt-get install ne
```

to install the program `ne`

Works on Ubuntu and Debian (and derivatives)

Exercise 1

Build an image (i.e. write the Dockerfile) based on the ubuntu:18.04 image and runs the command

```
df -h
```

saving the output in a directory outside the container.

```
docker run --rm -v $HOME/data:/data elixir1
```

can be used to run the container

What are the permission of the new file?

Solution

Installing programs

1. Ubuntu/Debian packages list or `apt search`
2. Conda packages. search or `conda search`
3. Linuxbrew. List or `brew search`
4. From specialized archives: e.g. `pip`
5. Compile the program

Ubuntu packages

- Lots of packages ready `apt-get install ne`
- Very robust
- Additional packages with PPAs

  `add-apt-repository ppa:user/ppa-name`
  - Not very useful for science

Advantages

- Safe and efficient
- Easy to rebuild from source

Cons

- Sometimes outdated (Ubuntu LTS are released every 2 years)

Conda packages

- Must be downloaded and installed (with bash)
- Relies on channels
  - *base* with installation
  - *bioconda*: bioinformatics programs
  - *conda-forge*: misc programs
- Not very robust/stable

Advantages

- Programs are usually the latest version

Cons

- Needs update to config files (`~/.bashrc`)
- Weird interactions with system programs/libraries
- Messes up with `$PATH` (e.g. shebang)

Linuxbrew packages

- Must be downloaded and installed
- Only one source of packages
- Similar to conda
  - fewer packages

Exercise 2

Build an image (i.e. write the Dockerfile) that:

1. installs the 1.7 version of `samtools` (via `apt-get`)
2. runs `samtools` saving the output in a directory outside the container.

```
docker run --rm -v $HOME/data:/data elixir2
```

can be used to run the container

Solution

Exercise 3

Build an image (i.e. write the Dockerfile) that:

1. downloads the version 1.9 of `samtools`
2. compiles `samtools`
3. runs `samtools` saving the output in a directory outside the container.

```
docker run --rm -v $HOME/data:/data elixir3
```

can be used to run the container

Solutions Ex. 3

Development version

Production version

In production we want to minimize the *image size*, in development we want to minimize *runtime* (exploit the cache)

Exercise 4

Build an image (i.e. write the Dockerfile) that:

1. runs a version of `samtools` that has been stored and compiled outside the container (e.g. in `$HOME/code/`), saving the output in a directory outside the 3 container.
2. find the best `docker run` invokation

Solution Ex. 4

Solution

Exercise 5

Build an image (i.e. write the Dockerfile) that:

1. uses `conda` to install the latest version of `samtools`
2. runs `samtools` saving the output in a directory outside the container.

```
docker run --rm -v $HOME/data:/data elixir5
```

can be used to run the container

Solution Ex. 5

Solution