Docker

What it can do and why you would want to use it

(Some) History

- OS-level virtualization on Linux
  - chroot (1982)
    - just changes apparent root directory
    - a number of limitations (stacking, jail break)
  - Linux-VServer (2001)
    - uses security contexts provided by kernel
    - jail mechanism
  - LXC (Linux Containers, 2008)
    - uses kernel cgroups/namespace isolation
  - Docker (2013)
    - uses libcontainer, libvirt, LXC, systemd-nspawn
    - containers are standard processes

Docker Components

- Software
  - dockerd: daemon, manages containers
  - docker: CLI for dockerd
- Objects
  - image: read-only template for containers
  - container: environment that runs applications
  - service: for scaling across multiple dockerd (aka swarm)
- Registries
  - repository for images (pull/push)
  - public or private
  - main public repo: Docker Hub
  - other backends, e.g.: sonatype nexus repository

Source: Wikipedia

Installation

- CLI

  sudo apt install docker.io

- Docker Desktop (check terms re commercial use!)

  https://docs.docker.com/desktop/install/linux-install/

Using images

- Pull image

  docker pull <URL>

  docker pull hello-world
- URL format

  [registry-url/]namespace/image[:tag]
- "Official" images have no namespace (implicit "library"), hence only "hello-world"

Manage images

- Sub-command "image"
- list images: docker image ls
- inspect image: docker image inspect ...
- delete image(s): docker image rm ...
- remove unused images: docker image prune ...

Using images (2)

- Spin up container from image

  docker run <URL>

  docker run hello-world

- Output

  Hello from Docker!

  This message shows that your installation appears to be working correctly.

  ...

Using images (3)

- Two types of volumes
  - named volumes: persistent, eg for databases
  - mapped directories

    -v/--volume or --mount (greater control than -v)

    -v HOSTDIR:CONTAINERDIR

Managing Containers

- Sub-command "container"
- List running containers: docker container ls
- List all containers: docker container ls -a
- Start/Stop container: docker container start/stop ...
- Remove container(s): docker container rm ...

Create images

- Image specification: Dockerfile
- Basic commands:
  - FROM: the base image
  - RUN: executes command (use)
  - ENV: set persistent environment variable
  - COPY: copy files from host into image (from within context)
  - WORKDIR: sets the current working dir, creates it automatically
- Each command is a layer
- Docker detects changes in Dockerfile by hashing the commands (hence use versions!)
- Build image with sub-command "build" (same dir as Dockerfile, aka context)

  docker build -t TAGNAME .

Clean up

- stop all containers

  docker stop $(docker ps -a -q)
- remove all containers

  docker rm $(docker ps -a -q)
- purging all unused or dangling images, containers, volumes, and networks:

  docker system prune
- you can be even more aggressive when adding the -a flag:

  docker system prune -a

Why use it?

- Isolating applications
- Legacy software with outdated libraries
- Conflicting library requirements (eg CUDA)
- Easily deploy complex applications
- Version applications (easily switch versions)

If you want to know more...

- Introduction for Data Scientists

  https://www.data-mining.co.nz/docker-for-data-scientists/
- Docker reference

  https://docs.docker.com/engine/reference/builder/