Getting started with containers

Singularity & pipelines

Why singularity?

On a HPC:

- Users have different **levels** of **priviliges**
- Users **submit jobs** with time/cpu/memory restrictions

These are two things **Docker doesn’t facilitate**:

- Requires **superuser** priviliges
- Docker commands are an API of a **daemon** (parentless)

Specific singularity features

- UID + permissions are inside the container always the same as outside
- No daemon – container is a child process, which also means:
    - Images are files (.sif)
    - Different image format

Docker + singularity

- “Best of both worlds”
- Most bioinformaticians use `docker` for:
  - Development
  - testing - CI/CD
  - Sharing
- Most bioinformaticians use `singularity` for:
  - Deployment on a HPC

```
singularity pull docker://namespace/image:tag
```

Singularity without docker

- The singularity ‘Dockerfiles` : Singularity recipies (i.e. definition file)
- Singularity hub: https://singularity-hub.org/
- **Note**: installing as root inside container requires root privileges outside container!
- Solutions: --fakeroot or build with external runner

Pipeline development

- **Pipelines** need to be easily **reproducible** over all platforms
- **Containers** support that
- Most bioinformatic **tools** are available as a **container**, e.g.: bioconda.github.io

BIOCONDA®

Containers in pipeline

main.nf

```
process foo {
        container 'namespace/repo:tag'

        """
        my_command.sh
        """
}
```

Snakefile

```
rule foo:
        input: "input_file.fastq"
        output: "output_file.html"
        container: "docker://namespace/repo:tag"
        shell: "my_command.sh"
```

Pipeline development

- Both **snakemake** and **nextflow** support conda, docker and singularity
- **Containers** can be built based on **conda** environments
- Once your cluster environment is set: **ultimate scalability/reproducibility**

Snakemake example

`envs/ggplot.yaml`

```
channels:
-  r
dependencies:
- r=3.3.1
- r-ggplot2=2.1.0
```

`myrule.smk`

```
rule plot-stuff:
    input: "table.txt"
    output: "plots/myplot.pdf"
    conda: "envs/ggplot.yaml"
    script: "scripts/plot-stuff.R"
```

```
snakemake --containerize > Dockerfile
```

`myrule_containerized.smk`

```
containerized: "docker://username/myworkflow:1.0.0"

rule plot-stuff:
    input: "table.txt"
    output: "plots/myplot.pdf"
    conda: "envs/ggplot.yaml"
    script: "scripts/plot-stuff.R"
```

https://snakemake.readthedocs.io/en/stable/snakefiles/deployment.html#containerization-of-conda-based-workflows

Advantages

- Specify your environment once (in the yaml) and:
  - Run using conda
  - Run using docker/singularity (platform independent)
- Improve **readability** (conda yaml)
- **No** need to **re-**download and **re-**install conda dependencies if **re-**running pipeline inside container

Exercises

- Pull your own docker image with singularity
- Container execution and mounting with singularity
- Using a biocontainers image to do some bioinformatics