Portable Workflows II

CMSE 890-402

Portable workflows: Scripting

Considerations:

- Dependencies (interpreter version)
- Environments (conda, pip)
- Containers

Dependencies

- Ensure the minimum and *maximum* compatible versions are listed
- Provide lists of required packages
- Provide installation instructions for compatible operating systems

Environments

- Include either:
  - requirements.txt, a list of pip packages that are required, for pure Python workflows
  - Environment YAML file for conda
- Note that the environment YAML can include:
  - Conda packages
  - Pip packages (in a separate block)
  - R packages (for an R environment)

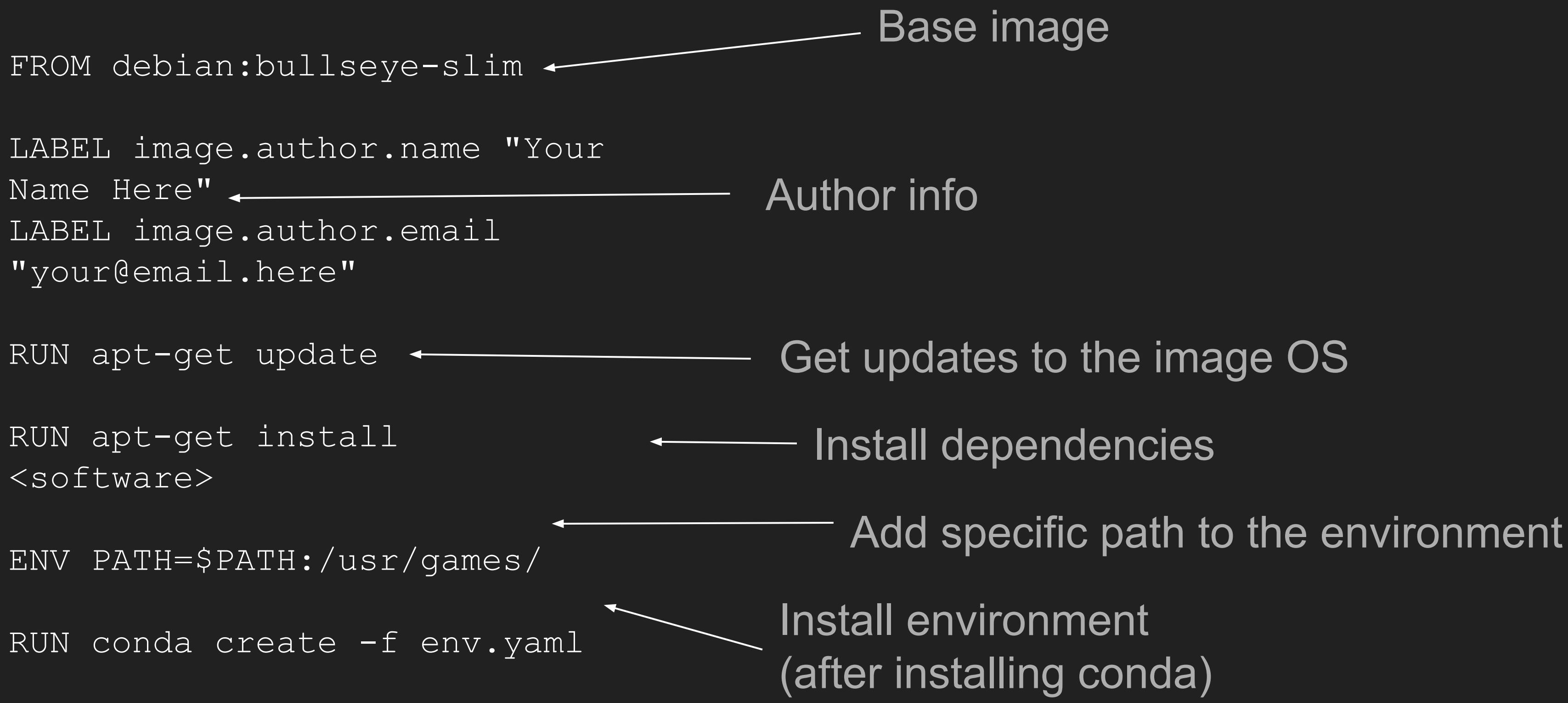
Containers

1. Choose a compatible base container OS
2. Use that to install dependencies
3. Install the environment into the container

These steps can apply to all dependencies of the workflow too!

REMINDER: Singularity will automatically connect useful local paths, Docker *does not*. NextFlow handles the Docker paths automatically.

Container setup: Docker



```
FROM debian:bullseye-slim

LABEL image.author.name "Your
Name Here"
LABEL image.author.email
"your@email.here"

RUN apt-get update

RUN apt-get install
<software>

ENV PATH=$PATH:/usr/games/

RUN conda create -f env.yaml
```

Base image

Author info

Get updates to the image OS

Install dependencies

Add specific path to the environment

Install environment
(after installing conda)

Container setup: Singularity

```
Bootstrap: docker
From: debian:bullseye-slim

%environment
export PATH=$PATH:/usr/games/

%labels
AUTHOR <your name>

%post

apt-get update

apt-get install <software>

conda create -f env.yaml
```

Use dockerhub

Base image OS

Add specific path to the environment

Author info

Get updates to the image OS

Install dependencies

Install environment
(after installing conda)

Portable workflows: SnakeMake

- SnakeMake has a recommended directory structure to follow
  - Git repository template available! https://github.com/snakemake-workflows/snakemake-workflow-template/generate
- SnakeMake supports conda environments per rule
  - Environments can be frozen with specific package choices
- Use containers with `--use-singularity` (no global config override!)
  - Container image is referenced under the “container” rule directive
  - No direct Docker support, all via Singularity!
- SnakeMake can *automatically* generate a container with

```
--containerize > Dockerfile
```

  - This contains all necessary environments
  - Used with the global directive “containerized”

Portable workflows: NextFlow

- NextFlow supports conda and containers
- Uses different command line options for Docker and Singularity
- Typically prefers a single container with all dependencies
- Can combine a container with a conda env like SnakeMake
  - Requires manual Dockerfile creation
- As with SnakeMake, can apply a container to every process or per-process
  - Global: use NextFlow config file `process.container`
  - Per-process: use process directive `container`
  - Need to set the container type in the config with `docker.enabled` or `singularity.enabled = true`

Many containers or one container?

Many

Pros:

- Individual software easy to update
- Reusable across other workflows
- Easy to parallelize

Cons:

- Many files
- Possible repeated data

One

Pros:

- Just one file
- Easy setup
- No repetition

Cons:

- Hard to parallelize
- Hard to update without rebuilding entire workflow

In-class assignment

Fill out this feedback form:

https://forms.gle/3Q16CQ3BayyD4fxSA