Theiagen®

GENOMICS

We will be starting soon

Docker for Public Health Bioinformatics

Week 03 - Developing Custom Docker Images

PRESENTED BY:
Inês Mendes, PhD

A Mid-Atlantic Workforce Development Offering Provided by the Division of Consolidated Laboratory Services in Collaboration with Theiagen Genomics

Course Introduction

Training Workshop Resources

Training Information, Communication, and Support

- **GitHub Repository** created to host training resources and information:
  - https://github.com/theiagen/Mid-Atlantic-Docker4PH-2025
- **Support contact:**
  - support@theiagen.com

Theiagen® GENOMICS

Course Agenda

Docker for Public Health Bioinformatics

Week 3 - April 15/17, 2025

- Developing Custom Docker Images
- Hands-on Exercise: Building and Sharing a Custom Dockerfile

Theiagen GENOMICS

OBJECTIVE

Goals by End of Week 3

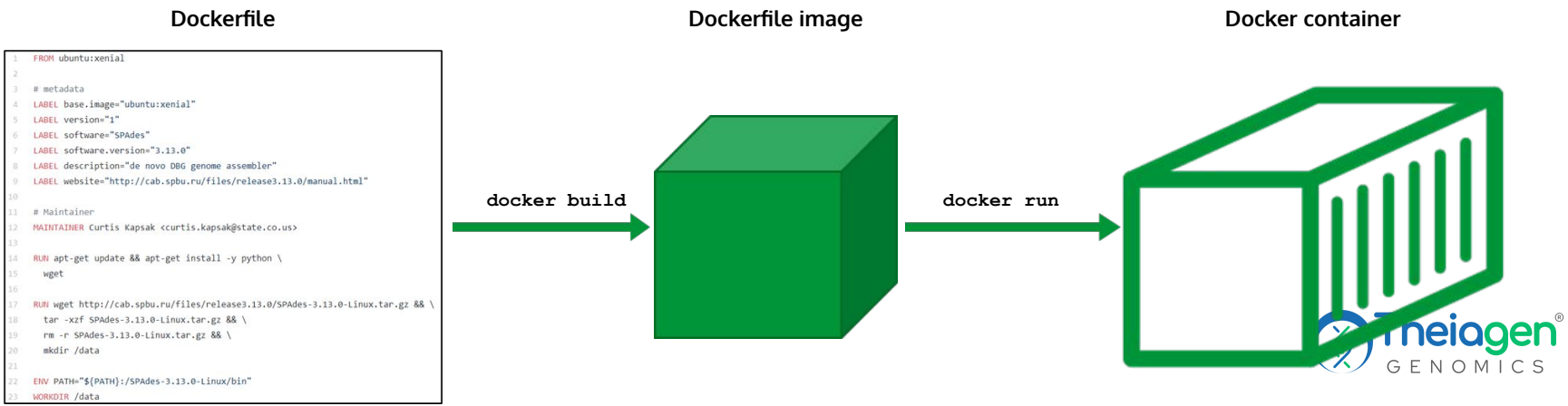
- Learn best-practices for developing and testing Dockerfiles
- Learn strategies for creating new Dockerfiles for bioinformatics software
- Gain experience developing and testing a Dockerfile

Week 1 & 2 Review

Week 1 Review

Summary:

- **Dockerfile** is used to create the docker **image**
- Docker **image** is used to create the docker **container**
  - Container **is the runnable instance of an image**



Week 1 review

Docker Images can be built locally **or** pre-built images can be downloaded from public repositories like:

- **Docker hub:** https://hub.docker.com/
- **Quay.io:** https://quay.io/
- **GitHub container registry (GHCR):** https://ghcr.io/
- Cloud provider container registries:
  - GCP Artifact Registry
  - Amazon Elastic Container Registry
  - Microsoft Azure Container Registry
- Private registries are an (paid) option

Theiagen® GENOMICS

Week 1 Review

- **Docker Hub:** https://hub.docker.com
- **Quay.io:** https://quay.io/

Theiagen GENOMICS

Week 2 Review

Dockerfile instructions

- `FROM` defines the base docker image
- `ARG` set environmental variables ONLY available during build time
- `ENV` set environmental variables that persist during and after build time
- `RUN` executes a command in a new layer
- `WORKDIR` sets the working directory for executing commands
- `COPY` (and `ADD`) copy files into the docker image
- `LABEL` adds metadata to your docker image

Theiagen® GENOMICS

Week 2 Review

Docker build

- Builds an image from a Dockerfile
- At a minimum, requires a Dockerfile. Some dockerfiles require other files for building (scripts, databases, etc.)
- Official docs: https://docs.docker.com/engine/reference/commandline/build/

```
docker build --tag <name>:<tag> <directory-with-dockerfile>
```

```
docker build --tag spades:3.15.5 spades/3.15.5/
```

Theiagen GENOMICS

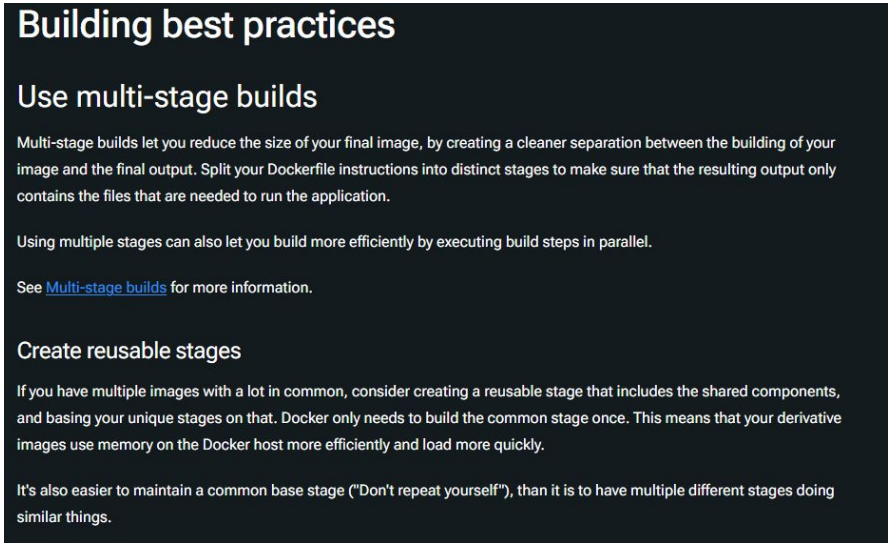
Developing Custom Docker Images

Best Practises for Writing Dockerfiles

- One docker container should be used for one purpose - one bioinfo tool*
  - * There are some exceptions!
- Only install what is necessary. Avoid installing extra programs to keep disk usage low
- Fewer layers = better. `RUN`, `COPY`, and `ADD` instructions add layers
- Readability of Dockerfile is helpful. Usually use one command per line
- No “large” databases or files. Large means >1GB
  - There are exceptions, but usually it’s better practice to bring large databases into the container at runtime instead of keeping in container

Theiagen® GENOMICS

Best Practises for Writing Dockerfiles



Building best practices

Use multi-stage builds

Multi-stage builds let you reduce the size of your final image, by creating a cleaner separation between the building of your image and the final output. Split your Dockerfile instructions into distinct stages to make sure that the resulting output only contains the files that are needed to run the application.

Using multiple stages can also let you build more efficiently by executing build steps in parallel.

See Multi-stage builds for more information.

Create reusable stages

If you have multiple images with a lot in common, consider creating a reusable stage that includes the shared components, and basing your unique stages on that. Docker only needs to build the common stage once. This means that your derivative images use memory on the Docker host more efficiently and load more quickly.

It's also easier to maintain a common base stage ("Don't repeat yourself"), than it is to have multiple different stages doing similar things.

https://docs.docker.com/develop/develop-images/dockerfile_best-practices/

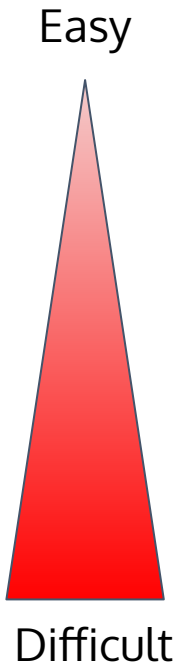
Theiagen® GENOMICS

Strategies for Creating Dockerfiles

General tips

- **`docker build`** often while writing Dockerfile. Trial and error as much as necessary!
- If looking for the location of files, launch interactive container to see where files are located: **`docker run -it <image>`**
  - alternatively - add **`ls`**, **`find`**, or other commands in your dockerfile
- Helpful to have the Dockerfile open in front of you when building an image. VSCode makes it easy for us
- Use a Dockerfile linter (such as the Docker VSCode extension) to catch errors before you **`docker build`**
- Use **`docker build --progress=plain`** so that all STDOUT/STDERR is printed to screen - can see every command being executed

Strategies for Creating Dockerfiles



I want to create a dockerfile, where do I start?

- Easiest - Use & modify an existing dockerfile
- Tool developers (or tool users) may provide their own dockerfiles
    - If the dockerfile code is open source (and licensed as such) it's fair to use with proper attribution!
- A bit more challenging - start from a template dockerfile
- Most challenging - writing a dockerfile from scratch

Strategies for Creating Dockerfiles

I'm starting from a template or pre-existing dockerfile, where do I start?

1. Read the tool's documentation. Familiarize yourself with the installation procedure. How does the tool author recommend to install the tool?
2. See what programming language the tool is written in - that will dictate how things are installed. Python, Perl, Rust, R, C/C++, something else?
    a. Does the installation require code compilation?
    b. If so, does the tool author provide pre-compiled binaries (executables)?
        i. pre-compiled binaries are usually easier to download and use than compiling code as part of Dockerfile

Theiagen® GENOMICS

Strategies for Creating Dockerfiles

I'm starting from a template or pre-existing dockerfile, where do I start?

1. Remember that not every user will be running the container as the `root` linux user. Singularity and other container engines may run containers as non-root users
    a. Make sure that required files (scripts, databases, etc. files) are readable and executable to all users. You may have to use **`chmod`** command to change permissions on files

Theiagen® GENOMICS

Strategies for Creating Dockerfiles

I'm starting from a template or pre-existing dockerfile, where do I start?

2. Place files in an expected location & document where important files are located.
    a. example: Mummer docker image used for ANI for enteric pathogens
        i. https://github.com/theiagen/docker-builds/tree/master/mummer/4.0.0-RGDv2
        ii. *"The FASTA files for RGDv2 can be found within the directory /RGDv2/ inside the docker image."*

Theiagen® GENOMICS

Strategies for Creating Dockerfiles

Programming language specific tips

Python

- first - install python and try installing python dependencies (e.g. **`numpy`**) **`via apt-get`**
- second - install **`pip`** using **`apt-get`**, then use **`pip`** to install specific python packages
    - advantage: easy to pin versions
- example: NanoPlot

```
16  # install dependencies via apt; cleanup apt garbage; set locale to en_US.UTF-8
17  RUN apt-get update && apt-get install -y zlib1g-dev \
18   bzip2 \
19   libbz2-dev \
20   liblzma-dev \
21   libcurl4-gnutls-dev \
22   libncurses5-dev \
23   libssl-dev \
24   python3 \
25   python3-pip \
26   python3-setuptools \
27   locales && \
28   locale-gen en_US.UTF-8 && \
29   apt-get autoclean && rm -rf /var/lib/apt/lists/*
30
31  # for singularity compatibility
32  ENV LC_ALL=C
33
34  # install NanoPlot via pypi using pip3; make /data directory
35  RUN pip3 install matplotlib psutil requests NanoPlot==${NANOPLOT_VER} && \
36   mkdir /data
```

Theiagen GENOMICS

Strategies for Creating Dockerfiles

Programming language specific tips

Perl

- first - try installing perl dependencies (e.g. DateTime) via **`apt-get`**
- second - install **`cpanm`** using **`apt-get`**, then use **`cpanm`** to install specific perl dependencies
- example: [Prokka]

```
28  # install dependencies
29  RUN apt-get update && apt-get -y --no-install-recommends install \
30   bzip2 \
31   gzip \
32   wget \
33   perl \
34   less \
35   libdatetime-perl \
36   libxml-simple-perl \
37   libdigest-md5-perl \
38   default-jre \
39   bioperl \
40   hmmer \
41   zlib1g-dev \
42   python \
43   liblzma-dev \
44   libbz2-dev \
45   xz-utils \
46   curl \
47   g++ \
48   cpanminus \
```

Theiagen GENOMICS

Strategies for Creating Dockerfiles

Programming language specific tips

Compiled languages (C, C++, Rust)

- Pre-compiled binaries
  - Usually are operating system or CPU architecture specific
  - You usually want the **64-bit Linux binaries. AKA x86_64**
- example: Mash

- When binaries are not available, you may have to compile the code yourself
  - May require **`gcc`** (C code) or **`g++`** (C++ code) for compiling the code
  - Other dependencies might also be required for compilation, usually tool authors will list those. Example: **`zlib1g-dev`**, **`make`**, etc.
- example: Samtools

Theiagen GENOMICS

Hands-On Exercise

Exercise 03: Building and Sharing a Custom Dockerfile

Exercise Goal:

- Access development environment via GitPod and VS Code
- [Optional] Create branch for Week 03
- Choose a tool to containerize
- Create custom Dockerfile
- Test and build custom Dockerfile
- [Extra content] Pushing to a remote Docker repository (https://hub.docker.com)

Theiagen® GENOMICS

Post-Training Feedback Form

Post-Training Feedback Form

- **Anonymous feedback form** to evaluate course delivery:
  - https://forms.gle/q55XabtYLhjpHCPf9
- **Support materials:**
  - **GitHub Repository** created to host training resources and information:
    - https://github.com/theiagen/Mid-Atlantic-Docker4PH-2025
  - **Support contact:**
    - support@theiagen.com

Theiagen® GENOMICS

Theiagen®

GENOMICS

www.theiagen.com
support@theiagen.com