Theiagen®
GENOMICS

We will be starting soon

Docker for Public Health Bioinformatics

Week 02 - Container Repositories and Writing Dockerfiles

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 2 - April 08/10, 2025

- Container Repositories and Writing Dockerfiles
- Hands-on Exercise: Writing Dockerfiles to Build Docker Images

Theiagen GENOMICS

OBJECTIVE

Goals by End of Week 2

- Learn about publicly available container registries & resources
- Understand the Dockerfile and how it is used for building docker images
- Learn best-practices for writing Dockerfiles
- Learn how to build a docker image on the command line using pre-defined Dockerfiles

Week 1 Review

Week 1 Review

Summary:

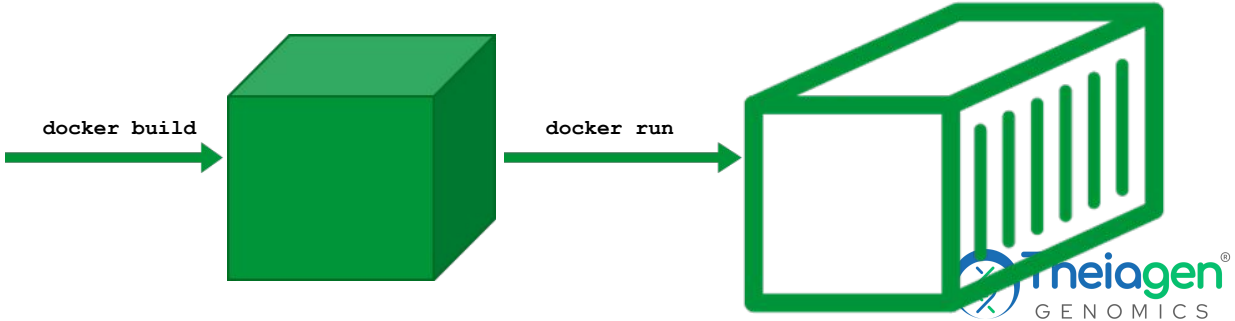
- **Dockerfile** is used to create the docker **image**
- Docker **image** is used to create the docker **container**
  - Container **is the runnable instance of an image**

Dockerfile

```
FROM ubuntu:xenial

# metadata
LABEL base.image="ubuntu:xenial"
LABEL version="1"
LABEL software="SPAdes"
LABEL software.version="3.13.0"
LABEL description="de novo DBG genome assembler"
LABEL website="http://cab.spbu.ru/files/release3.13.0/manual.html"

# Maintainer
MAINTAINER Curtis Kapsak <curtis.kapsak@state.co.us>

RUN apt-get update && apt-get install -y python \
  wget

RUN wget http://cab.spbu.ru/files/release3.13.0/SPAdes-3.13.0-Linux.tar.gz && \
  tar -xzf SPAdes-3.13.0-Linux.tar.gz && \
  rm -r SPAdes-3.13.0-Linux.tar.gz && \
  mkdir /data

ENV PATH="${PATH}:/SPAdes-3.13.0-Linux/bin"
WORKDIR /data
```

Dockerfile image

Docker container



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

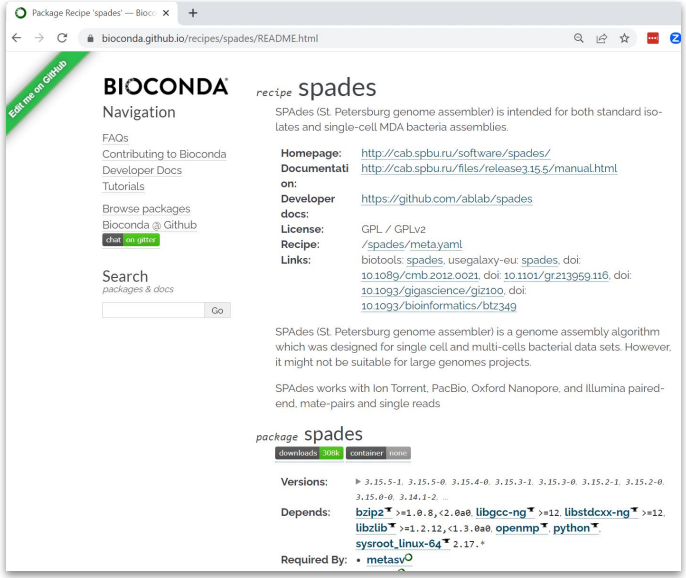
Week 1 Review

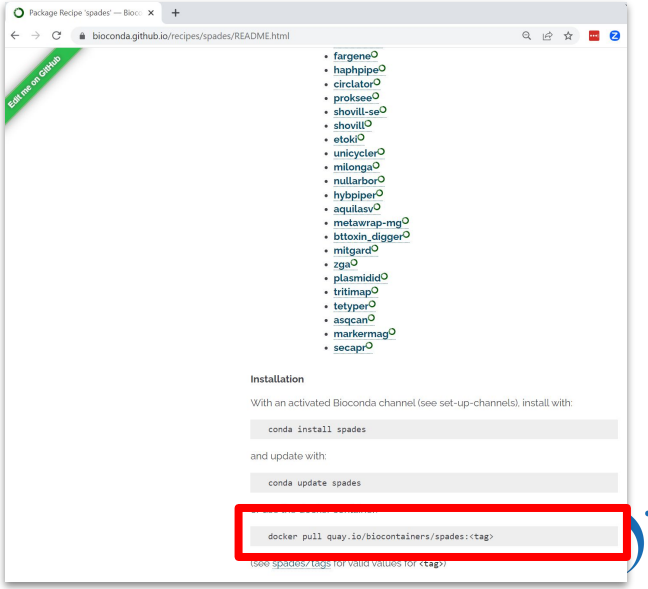
- **Docker Hub:** https://hub.docker.com
- **Quay.io:** https://quay.io/

Theiagen GENOMICS

Week 1 Review

- All bioconda packages are available as docker images on quay.io:
  - https://bioconda.github.io/recipes/spades/README.html





Theiagen® GENOMICS

Why Write my own Dockerfile?

Why Write my own Dockerfile?

- Not all docker images were created equally
  - Some (like biocontainers) are made by robots! 🤖
- Not all docker images work "out-of-the-box"
  - Limited-to-no testing performed with docker image
- "I cannot find a docker image for the software I want to use"
- "I want to know EXACTLY how X software was installed"
  - "Were versions pinned?"
- I want to include multiple tools in a single docker image"
  - minimap2 + samtools commands piped together:

```
minimap2 -x map-ont -a ref.fasta reads.fastq.gz | samtools sort -o out.bam
```

Getting Started with Dockerfiles

Review - The Dockerfile

- In order to build a docker image, you need at a minimum one file: **the `Dockerfile`**
  - **`Dockerfile`** = set of instructions used to build a docker image
  - Similar to an installation script or a **.yml** file used for making/sharing conda environments

Format

Here is the format of the `Dockerfile` :

```
# Comment
INSTRUCTION arguments
```

```
FROM
ubuntu:jammy

# install stuff!
RUN install \
    softwareA \
    softwareB \
    softwareC
```

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

The Dockerfile

- Dockerfile instructions (**FROM**, **RUN**, **COPY**, **ENV**, etc.) will add a "layer" to the docker image
- Images are multi-layered and different images may share layers like the base image
  - **FROM ubuntu:focal**

Spades Dockerfile

- https://github.com/StaPH-B/docker-builds/blob/master/spades/3.15.5/Dockerfile

Dockerfile Cheat Sheet

- https://kapeli.com/cheat_sheets/Dockerfile.docset/Contents/Resources/Documents/index

```
FROM ubuntu:focal as app

# to make it easier to upgrade for new versions; ARG variables only persist during docker image build time
ARG SPADES_VER="3.15.5"

LABEL base.image="ubuntu:focal"
LABEL dockerfile.version="2"
LABEL software="SPAdes"
LABEL software.version="${SPADES_VER}"
LABEL description="de novo DBG genome assembler"
LABEL website="https://github.com/ablab/spades"
LABEL license="https://github.com/ablab/spades/blob/v3.15.3/assembler/LICENSE"
LABEL maintainer="Curtis Kapsak"
LABEL maintainer.email="kapsakcj@gmail.com"

# install dependencies; cleanup apt garbage
# python v3.8.10 is installed here; point 'python' to python3
RUN apt-get update && apt-get install --no-install-recommends -y python3 \
 python3-distutils \
 wget \
 pigz \
 ca-certificates && \
 apt-get autoclean && rm -rf /var/lib/apt/lists/* && \
 update-alternatives --install /usr/bin/python python /usr/bin/python3 10

# install SPAdes binary; make /data
RUN wget http://cab.spbu.ru/files/release${SPADES_VER}/SPAdes-${SPADES_VER}-Linux.tar.gz && \
  tar -xzf SPAdes-${SPADES_VER}-Linux.tar.gz && \
  rm -r SPAdes-${SPADES_VER}-Linux.tar.gz && \
  mkdir /data

# set PATH and locale settings for singularity
ENV LC_ALL=C.UTF-8 \
    PATH="${PATH}:/SPAdes-${SPADES_VER}-Linux/bin"

WORKDIR /data

# test layer
FROM app as test

# print version and run the supplied test flag
RUN spades.py --version && spades.py --test && spades.py --help
```

The Dockerfile - FROM

- Initializes a new build stage
- Required - A valid Dockerfile must start with a **FROM** instruction
  - The only instruction that can precede **FROM** is an **ARG** variable (more on this later)
- **FROM** defines the base image
  - Recommendation - choose a base image and stick with it
- Official docs: https://docs.docker.com/engine/reference/builder/#from

FROM

```
FROM [--platform=<platform>] <image> [AS <name>]
```

Or

```
FROM [--platform=<platform>] <image>[:<tag>] [AS <name>]
```

Or

```
FROM [--platform=<platform>] <image>[@<digest>] [AS <name>]
```

master docker-builds / spades / 3.15.5 / Dockerfile

kapsakcj added ca-certificates to spades 3.15.5 dockerfile ✓

Code Blame 42 lines (34 loc) · 1.39 KB

```
1 FROM ubuntu:focal as app
```

Theiagen GENOMICS

The Dockerfile - FROM

My favorite base images

Ubuntu - hub.docker.com/_/ubuntu

- Familiar linux OS - has many basic linux commands installed (**`ls`**, **`cd`**, **`cp`**, **`mv`**, **`ps`**, etc.)
- Relatively easy to install dependencies via **`apt-get`**
  - packages.ubuntu.com for looking up what is available via **`apt-get`**
- Can use **`pip`** for python packages & **`cpan`**/**`cpanm`** for perl packages

```
FROM ubuntu:focal as app

# to make it easier to upgrade for new versions; ARG variables only persist during docker image build time
ARG SPADES_VER="3.15.5"

LABEL base.image="ubuntu:focal"
LABEL dockerfile.version="2"
LABEL software="SPAdes"
LABEL software.version="${SPADES_VER}"
LABEL description="de novo DBG genome assembler"
LABEL website="https://github.com/ablab/spades"
LABEL license="https://github.com/ablab/spades/blob/v3.15.3/assembler/LICENSE"
LABEL maintainer="Curtis Kapsak"
LABEL maintainer.email="kapsakcj@gmail.com"

# install dependencies; cleanup apt garbage
# python v3.8.10 is installed here; point 'python' to python3
RUN apt-get update && apt-get install --no-install-recommends -y python3 \
 python3-distutils \
 wget \
 pigz \
 ca-certificates && \
 apt-get autoclean && rm -rf /var/lib/apt/lists/* && \
 update-alternatives --install /usr/bin/python python /usr/bin/python3 10

# install SPAdes binary; make /data
RUN wget http://cab.spbu.ru/files/release${SPADES_VER}/SPAdes-${SPADES_VER}-Linux.tar.gz && \
  tar -xzf SPAdes-${SPADES_VER}-Linux.tar.gz && \
  rm -r SPAdes-${SPADES_VER}-Linux.tar.gz && \
  mkdir /data

# set PATH and locale settings for singularity
ENV LC_ALL=C.UTF-8 \
    PATH="${PATH}:/SPAdes-${SPADES_VER}-Linux/bin"

WORKDIR /data

# test layer
FROM app as test

# print version and run the supplied test flag
RUN spades.py --version && spades.py --test && spades.py --help
```

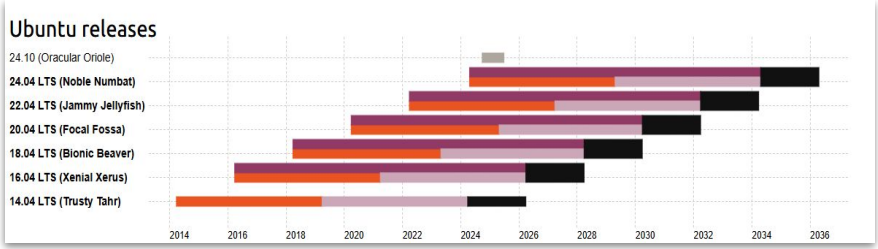
Theiagen® GENOMICS

The Dockerfile - FROM

My favorite base images

Ubuntu - hub.docker.com/_/ubuntu

- Recommendation
  - Ubuntu 22.04 LTS (jammy)
    - `FROM ubuntu:jammy`
  - Ubuntu 24.04 LST (noble)
    - `FROM ubuntu:noble`
- Older Ubuntu releases are nearing the end of support, use something new that will be supported long term!



https://ubuntu.com/about/release-cycle#ubuntu

Theiagen GENOMICS

The Dockerfile - FROM

My favorite base images

micromamba - hub.docker.com/r/mambaorg/micromamba

- Familiar linux OS (Debian)
- **`micromamba`** is preinstalled
  - **`micromamba`** is even more lightweight than **`conda`** or **`miniconda`**

master docker-builds / freyja / 1.4.2 / Dockerfile

kapsakcj added dockerfile in preparation for freyja 1.4.2 version release c

Code Blame 70 lines (57 loc) · 2.82 KB

```
FROM mambaorg/micromamba:1.4.1 as app

# Version arguments
# ARG variables only persist during build time
ARG FREYJA_SOFTWARE_VERSION="1.4.2"
```

```
# Create Freyja conda environment called freyja-env from bioconda recipe
# clean up conda garbage
RUN micromamba create -n freyja-env -c conda-forge -c bioconda -c defaults freyja=${FREYJA_SOFTWARE_VERSION} && \
 micromamba clean -a -y
```

- I use for complicated installations where I rely upon the conda package in bioconda ← not best practice!
- Also use for scenarios where I want to use a conda recipe file (.yml) for installation

Theiagen GENOMICS

The Dockerfile - ARG

- Sets environmental variables that are ONLY available during docker image build time
- Once image is built, all **ARG** variables are unset/removed
- Useful for specifying versions of tools to install; can make it easy to upgrade versions by only changing one line of code

master docker-builds / freyja / 1.4.2 / Dockerfile

kapsakcj added dockerfile in preparation for freyja 1.4.2 version release c

Code Blame 70 lines (57 loc) · 2.82 KB

```
FROM mambaorg/micromamba:1.4.1 as app

# Version arguments
# ARG variables only persist during build time
ARG FREYJA_SOFTWARE_VERSION="1.4.2"
```

```
# Create Freyja conda environment called freyja-env from bioconda recipe
# clean up conda garbage
RUN micromamba create -n freyja-env -c conda-forge -c bioconda -c defaults freyja=${FREYJA_SOFTWARE_VERSION} && \
 micromamba clean -a -y
```

The Dockerfile - ENV

- Sets permanent environmental variables that are available during image build and afterwards for all users
- Useful for setting the **$PATH** variable
- [Format:](#)

ENV

```
ENV <key>=<value> ...
```

```
# set PATH and locale settings for singularity
ENV LC_ALL=C.UTF-8 \
    PATH="${PATH}:/SPAdes-${SPADES_VER}-Linux/bin"
```

- Can set multiple variables in one **ENV** layer using line breaks with a backslash \
- Useful for tools that do not automatically get added to your **$PATH** variable

Theiagen GENOMICS

The Dockerfile - RUN

- Executes a command in a new layer
- Each **RUN** layer builds upon the previous **FROM** and **RUN** layers
- Changes made in **RUN** commands are saved in the final docker image
- Assume **/bin/sh** shell for running commands
- Docker official docs: https://docs.docker.com/engine/reference/builder/#run

RUN

RUN has 2 forms:

- `RUN <command>` (*shell* form, the command is run in a shell, which by default is `/bin/sh -c` on Linux or `cmd /S /C` on Windows)
- `RUN ["executable", "param1", "param2"]` (*exec* form)

The `RUN` instruction will execute any commands in a new layer on top of the current image and commit the results. The resulting committed image will be used for the next step in the `Dockerfile`.

Theiagen® GENOMICS

The Dockerfile - RUN

- Look at a Dockerfile in-depth: SPAdes
- Tricks-of-the-trade:
  - **`apt-get update && apt-get install`** in a single **RUN** statement, done early in Dockerfile

```dockerfile
FROM ubuntu:focal as app

# to make it easier to upgrade for new versions; ARG variables only persist during docker image build time
ARG SPADES_VER="3.15.5"

LABEL base.image="ubuntu:focal"
LABEL dockerfile.version="2"
LABEL software="SPAdes"
LABEL software.version="${SPADES_VER}"
LABEL description="de novo DBG genome assembler"
LABEL website="https://github.com/ablab/spades"
LABEL license="https://github.com/ablab/spades/blob/v3.15.3/assembler/LICENSE"
LABEL maintainer="Curtis Kapsak"
LABEL maintainer.email="kapsakcj@gmail.com"

# install dependencies; cleanup apt garbage
# python v3.8.10 is installed here; point 'python' to python3
RUN apt-get update && apt-get install --no-install-recommends -y python3 \
 python3-distutils \
 wget \
 pigz \
 ca-certificates && \
 apt-get autoclean && rm -rf /var/lib/apt/lists/* && \
 update-alternatives --install /usr/bin/python python /usr/bin/python3 10

# install SPAdes binary; make /data
RUN wget http://cab.spbu.ru/files/release${SPADES_VER}/SPAdes-${SPADES_VER}-Linux.tar.gz && \
  tar -xzf SPAdes-${SPADES_VER}-Linux.tar.gz && \
  rm -r SPAdes-${SPADES_VER}-Linux.tar.gz && \
  mkdir /data

# set PATH and locale settings for singularity
ENV LC_ALL=C.UTF-8 \
    PATH="${PATH}:/SPAdes-${SPADES_VER}-Linux/bin"

WORKDIR /data

# test layer
FROM app as test

# print version and run the supplied test flag
RUN spades.py --version && spades.py --test && spades.py --help
```

Theiagen® GENOMICS

The Dockerfile - RUN

- Look at a Dockerfile in-depth: SPAdes
- Tricks-of-the-trade:
  - **`apt-get update && apt-get install`** in a single **RUN** statement, done early in Dockerfile
  - Use syntax appropriate for your build stage (shell in ubuntu). Imagine you are using the command line

```dockerfile
FROM ubuntu:focal as app

# to make it easier to upgrade for new versions; ARG variables only persist during docker image build time
ARG SPADES_VER="3.15.5"

LABEL base.image="ubuntu:focal"
LABEL dockerfile.version="2"
LABEL software="SPAdes"
LABEL software.version="${SPADES_VER}"
LABEL description="de novo DBG genome assembler"
LABEL website="https://github.com/ablab/spades"
LABEL license="https://github.com/ablab/spades/blob/v3.15.3/assembler/LICENSE"
LABEL maintainer="Curtis Kapsak"
LABEL maintainer.email="kapsakcj@gmail.com"

# install dependencies; cleanup apt garbage
# python v3.8.10 is installed here; point 'python' to python3
RUN apt-get update && apt-get install --no-install-recommends -y python3 \
 python3-distutils \
 wget \
 pigz \
 ca-certificates && \
 apt-get autoclean && rm -rf /var/lib/apt/lists/* && \
 update-alternatives --install /usr/bin/python python /usr/bin/python3 10

# install SPAdes binary; make /data
RUN wget http://cab.spbu.ru/files/release${SPADES_VER}/SPAdes-${SPADES_VER}-Linux.tar.gz && \
  tar -xzf SPAdes-${SPADES_VER}-Linux.tar.gz && \
  rm -r SPAdes-${SPADES_VER}-Linux.tar.gz && \
  mkdir /data

# set PATH and locale settings for singularity
ENV LC_ALL=C.UTF-8 \
    PATH="${PATH}:/SPAdes-${SPADES_VER}-Linux/bin"

WORKDIR /data

# test layer
FROM app as test

# print version and run the supplied test flag
RUN spades.py --version && spades.py --test && spades.py --help
```

Theiagen® GENOMICS

The Dockerfile - RUN

- Look at a Dockerfile in-depth: SPAdes
- Tricks-of-the-trade:
  - **`apt-get update && apt-get install`** in a single **RUN** statement, done early in Dockerfile
  - Use syntax appropriate for your build stage (shell in ubuntu). Imagine you are using the command line
  - \ (backslashes) are for line breaks (readability)

```
FROM ubuntu:focal as app

# to make it easier to upgrade for new versions; ARG variables only persist during docker image build time
ARG SPADES_VER="3.15.5"

LABEL base.image="ubuntu:focal"
LABEL dockerfile.version="2"
LABEL software="SPAdes"
LABEL software.version="${SPADES_VER}"
LABEL description="de novo DBG genome assembler"
LABEL website="https://github.com/ablab/spades"
LABEL license="https://github.com/ablab/spades/blob/v3.15.3/assembler/LICENSE"
LABEL maintainer="Curtis Kapsak"
LABEL maintainer.email="kapsakcj@gmail.com"

# install dependencies; cleanup apt garbage
# python v3.8.10 is installed here; point 'python' to python3
RUN apt-get update && apt-get install --no-install-recommends -y python3 \
 python3-distutils \
 wget \
 pigz \
 ca-certificates && \
 apt-get autoclean && rm -rf /var/lib/apt/lists/* && \
 update-alternatives --install /usr/bin/python python /usr/bin/python3 10

# install SPAdes binary; make /data
RUN wget http://cab.spbu.ru/files/release${SPADES_VER}/SPAdes-${SPADES_VER}-Linux.tar.gz && \
  tar -xzf SPAdes-${SPADES_VER}-Linux.tar.gz && \
  rm -r SPAdes-${SPADES_VER}-Linux.tar.gz && \
  mkdir /data

# set PATH and locale settings for singularity
ENV LC_ALL=C.UTF-8 \
    PATH="${PATH}:/SPAdes-${SPADES_VER}-Linux/bin"

WORKDIR /data

# test layer
FROM app as test

# print version and run the supplied test flag
RUN spades.py --version && spades.py --test && spades.py --help
```

Theiagen® GENOMICS

The Dockerfile - RUN

- Look at a Dockerfile in-depth: SPAdes
- Tricks-of-the-trade:
  - **&&** bash operator is used to create long one-liners so that multiple commands are run sequentially and are dependent on each command running successfully

```
FROM ubuntu:focal as app

# to make it easier to upgrade for new versions; ARG variables only persist during docker image build time
ARG SPADES_VER="3.15.5"

LABEL base.image="ubuntu:focal"
LABEL dockerfile.version="2"
LABEL software="SPAdes"
LABEL software.version="${SPADES_VER}"
LABEL description="de novo DBG genome assembler"
LABEL website="https://github.com/ablab/spades"
LABEL license="https://github.com/ablab/spades/blob/v3.15.3/assembler/LICENSE"
LABEL maintainer="Curtis Kapsak"
LABEL maintainer.email="kapsakcj@gmail.com"

# install dependencies; cleanup apt garbage
# python v3.8.10 is installed here; point 'python' to python3
RUN apt-get update && apt-get install --no-install-recommends -y python3 \
 python3-distutils \
 wget \
 pigz \
 ca-certificates && \
 apt-get autoclean && rm -rf /var/lib/apt/lists/* && \
 update-alternatives --install /usr/bin/python python /usr/bin/python3 10

# install SPAdes binary; make /data
RUN wget http://cab.spbu.ru/files/release${SPADES_VER}/SPAdes-${SPADES_VER}-Linux.tar.gz && \
  tar -xzf SPAdes-${SPADES_VER}-Linux.tar.gz && \
  rm -r SPAdes-${SPADES_VER}-Linux.tar.gz && \
  mkdir /data

# set PATH and locale settings for singularity
ENV LC_ALL=C.UTF-8 \
    PATH="${PATH}:/SPAdes-${SPADES_VER}-Linux/bin"

WORKDIR /data

# test layer
FROM app as test

# print version and run the supplied test flag
RUN spades.py --version && spades.py --test && spades.py --help
```

Theiagen® GENOMICS

The Dockerfile - RUN

- Look at a Dockerfile in-depth: SPAdes
- Tricks-of-the-trade:
  - **`&&`** bash operator is used to create long one-liners so that multiple commands are run sequentially and are dependent on each command running successfully
  - Assume **`/bin/sh`** shell for running commands, but there is a way to set **`/bin/bash`** as normal shell. Not necessary unless using bash-specific cmdline tricks
  -

```
FROM ubuntu:focal as app

# to make it easier to upgrade for new versions; ARG variables only persist during docker image build time
ARG SPADES_VER="3.15.5"

LABEL base.image="ubuntu:focal"
LABEL dockerfile.version="2"
LABEL software="SPAdes"
LABEL software.version="${SPADES_VER}"
LABEL description="de novo DBG genome assembler"
LABEL website="https://github.com/ablab/spades"
LABEL license="https://github.com/ablab/spades/blob/v3.15.3/assembler/LICENSE"
LABEL maintainer="Curtis Kapsak"
LABEL maintainer.email="kapsakcj@gmail.com"

# install dependencies; cleanup apt garbage
# python v3.8.10 is installed here; point 'python' to python3
RUN apt-get update && apt-get install --no-install-recommends -y python3 \
 python3-distutils \
 wget \
 pigz \
 ca-certificates && \
 apt-get autoclean && rm -rf /var/lib/apt/lists/* && \
 update-alternatives --install /usr/bin/python python /usr/bin/python3 10

# install SPAdes binary; make /data
RUN wget http://cab.spbu.ru/files/release${SPADES_VER}/SPAdes-${SPADES_VER}-Linux.tar.gz && \
  tar -xzf SPAdes-${SPADES_VER}-Linux.tar.gz && \
  rm -r SPAdes-${SPADES_VER}-Linux.tar.gz && \
  mkdir /data

# set PATH and locale settings for singularity
ENV LC_ALL=C.UTF-8 \
    PATH="${PATH}:/SPAdes-${SPADES_VER}-Linux/bin"

WORKDIR /data

# test layer
FROM app as test

# print version and run the supplied test flag
RUN spades.py --version && spades.py --test && spades.py --help
```

Theiagen® GENOMICS

The Dockerfile - WORKDIR

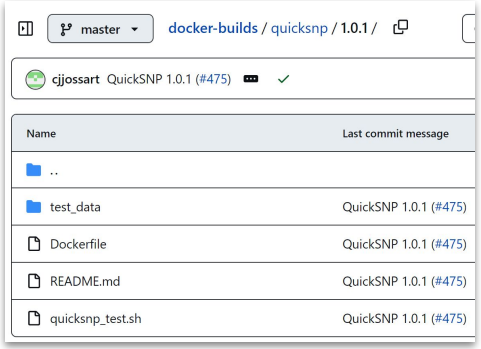
- Sets the working directory for any **RUN**, **CMD**, **ENTRYPOINT**, **COPY** and **ADD** instructions that follow it in the Dockerfile.
- **WORKDIR** will also persist after the image is build
- Can use multiple **WORKDIR**'s in one dockerfile, but only last **WORKDIR** will be saved in final image
- All StaPH-B/docker-builds images have the final **WORKDIR** set to **/data**

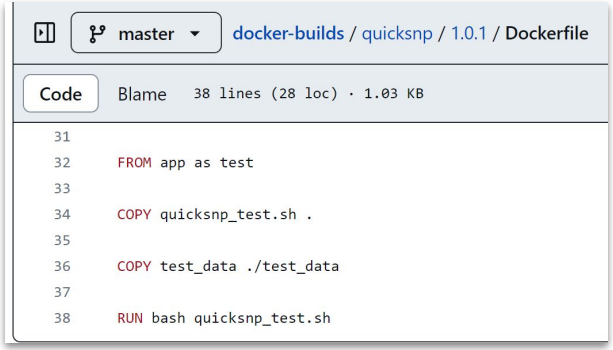
```
FROM ubuntu:focal as app

# to make it easier to upgrade for new versions; ARG variables only persist during docker image build time
ARG SPADES_VER="3.15.5"

LABEL base.image="ubuntu:focal"
LABEL dockerfile.version="2"
LABEL software="SPAdes"
LABEL software.version="${SPADES_VER}"
LABEL description="de novo DBG genome assembler"
LABEL website="https://github.com/ablab/spades"
LABEL license="https://github.com/ablab/spades/blob/v3.15.3/assembler/LICENSE"
LABEL maintainer="Curtis Kapsak"
LABEL maintainer.email="kapsakcj@gmail.com"

# install dependencies; cleanup apt garbage
# python v3.8.10 is installed here; point 'python' to python3
RUN apt-get update && apt-get install --no-install-recommends -y python3 \
 python3-distutils \
 wget \
 pigz \
 ca-certificates && \
 apt-get autoclean && rm -rf /var/lib/apt/lists/* && \
 update-alternatives --install /usr/bin/python python /usr/bin/python3 10

# install SPAdes binary; make /data
RUN wget http://cab.spbu.ru/files/release${SPADES_VER}/SPAdes-${SPADES_VER}-Linux.tar.gz && \
  tar -xzf SPAdes-${SPADES_VER}-Linux.tar.gz && \
  rm -r SPAdes-${SPADES_VER}-Linux.tar.gz && \
  mkdir /data

# set PATH and locale settings for singularity
ENV LC_ALL=C.UTF-8 \
    PATH="${PATH}:/SPAdes-${SPADES_VER}-Linux/bin"

WORKDIR /data

# test layer
FROM app as test

# print version and run the supplied test flag
RUN spades.py --version && spades.py --test && spades.py --help
```

Meldgen® GENOMICS

The Dockerfile - COPY

- Copies files into the image
- Files must be either:
  - located in same directory as Dockerfile (AKA the “build context”)
  - From another stage of the build process
- Useful for copying in test data, (small) databases, additional code:





Theiagen® GENOMICS

The Dockerfile - LABEL

- Optional but highly recommended!
- Allows addition of metadata to your docker image
- Generally located near top of dockerfile, after **FROM**
- See the template Dockerfile for examples:
  - https://github.com/theiagen/Mid-Atlantic-Docker4PH-2025/blob/main/docker/example/Dockerfile

master docker-builds / dockerfile-template / Dockerfile

SarahNadeau Update contribution documentation for testing (#280)

Code Blame 60 lines (48 loc) · 2.14 KB

```
# FROM defines the base docker image. This command has to come first in the file
# The 'as' keyword lets you name the folowing stage. We use `app` for the production image
FROM ubuntu:focal as app

# ARG sets environment variables during the build stage
ARG SOFTWARENAME_VER="1.0.0"

# LABEL instructions tag the image with metadata that might be important to the user
# Optional, but highly recommended
LABEL base.image="ubuntu:focal"
LABEL dockerfile.version="1"
LABEL software="SoftwareName"
LABEL software.version=$SOFTWARENAME_VER
LABEL description="This software does X, Y, AND Z!"
LABEL website="https://github.com/StaPH-B/docker-builds"
LABEL license="https://github.com/StaPH-B/docker-builds/blob/master/LICENSE"
LABEL maintainer="FirstName LastName"
LABEL maintainer.email="my.email@email.com"
```

Theiagen GENOMICS

The Dockerfile - comments

- Also optional but highly recommended!
- Comment lines begin with #

master docker-builds / dockerfile-template / Dockerfile

SarahNadeau Update contribution documentation for testing (#280)

Code Blame 60 lines (48 loc) · 2.14 KB

```
# FROM defines the base docker image. This command has to come first in the file
# The 'as' keyword lets you name the folowing stage. We use `app` for the production image
FROM ubuntu:focal as app

# ARG sets environment variables during the build stage
ARG SOFTWARENAME_VER="1.0.0"

# LABEL instructions tag the image with metadata that might be important to the user
# Optional, but highly recommended
LABEL base.image="ubuntu:focal"
LABEL dockerfile.version="1"
LABEL software="SoftwareName"
LABEL software.version=$SOFTWARENAME_VER
LABEL description="This software does X, Y, AND Z!"
LABEL website="https://github.com/StaPH-B/docker-builds"
LABEL license="https://github.com/StaPH-B/docker-builds/blob/master/LICENSE"
LABEL maintainer="FirstName LastName"
LABEL maintainer.email="my.email@email.com"

# RUN executes code during the build
# Install dependencies via apt-get or yum if using a centos or fedora base
RUN apt-get update && apt-get install -y \
 dependency_a \
 dependency_b \
 dependency_c

# Install and/or setup more things. Make /data for use as a working dir
# Example: ncbi-blast+ 2.9.0
RUN wget ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/2.9.0/ncbi-blast-2.9.0+-x64-linux.tar.gz && \
 tar -xzf ncbi-blast-2.9.0+-x64-linux.tar.gz && \
 rm ncbi-blast-2.9.0+-x64-linux.tar.gz && \
 mkdir /data
```

Theiagen® GENOMICS

Building docker images

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/
- General command structure:

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

- example using SPAdes dockerfile:

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

Theiagen GENOMICS

Multi-Stage Docker Builds

- Docker images that use multiple **FROM** instructions
  - can use a new base image **OR**
  - can use previous stage
- Useful for "optimizing" Dockerfiles:
  - reducing number of layers
  - removing unnecessary software & files, etc. to reduce final size of docker image

```
FROM baseimage as name
RUN stuff

FROM name as name2
RUN stuff

FROM name2 as final
RUN stuff
```

Theiagen GENOMICS

Multi-Stage Docker Builds

```
FROM ubuntu:focal as stage1
RUN stuff

FROM python:slim as stage2
RUN stuff

FROM ubuntu:jammy as finalstage
RUN stuff
```

Theiagen GENOMICS

Hands-On Exercise

Exercise 02: Writing Dockerfiles to Build Docker Images

Exercise Goal:

- Access development environment via GitPod and VS Code
- [Optional] Create branch for Week 02
- Building a Pre-Existing Dockerfile
- Editing a Pre-Existing Dockerfile
- [Extra content] Pushing to a remote Docker repository (https://hub.docker.com)

Theiagen GENOMICS

Theiagen®
GENOMICS

www.theiagen.com
support@theiagen.com