Docker for Reproducible Research

Anastasiia Enne

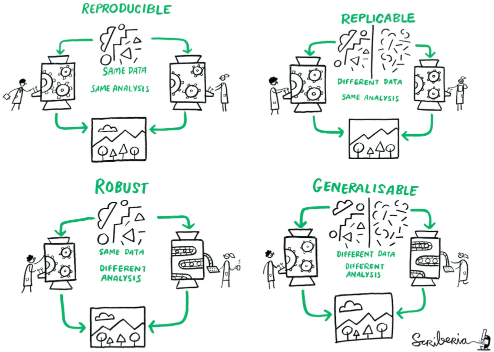
Group meeting, 14.01.2026

Overview

1. **Reproducibility crisis** — why just publishing your code is not enough
2. **What is Docker** and how can it help us with reproducibility
3. **Dockerizing an existing project** — a step-by-step guide if you want to try Docker

Reproducibility crisis

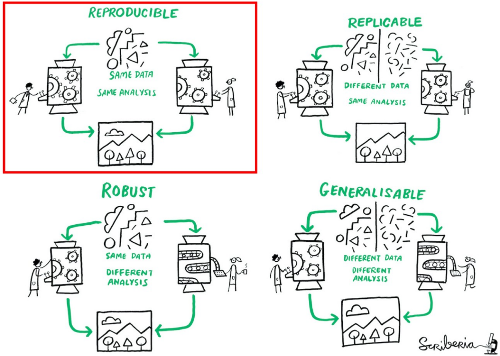
- Reminder: 4 definitions of reproducible research[1]



[1] The Turing Way Community. This illustration is created by Scriberia with The Turing Way community, used under a CC-BY 4.0 licence. DOI: 10.5281/zenodo.3332807

Reproducibility crisis

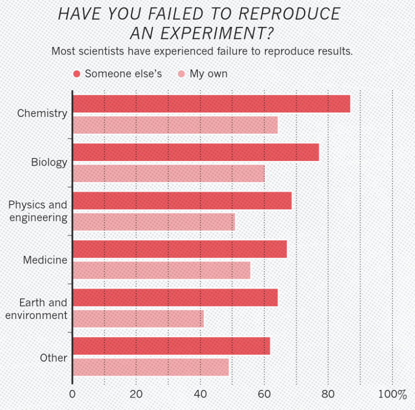
- Reminder: 4 definitions of reproducible research[1]
- Our focus will be on REPRODUCIBILITY



[1] The Turing Way Community. This illustration is created by Scriberia with The Turing Way community, used under a CC-BY 4.0 licence. DOI: 10.5281/zenodo.3332807

Reproducibility crisis

- Reminder: 4 definitions of reproducible research[1]
- Our focus will be on REPRODUCIBILITY
- Most scientists report that they have failed to reproduce an experiment[2]

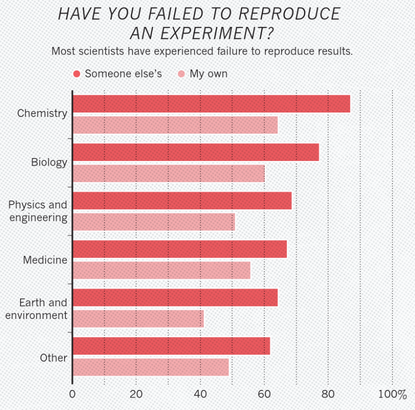


[1] The Turing Way Community. This illustration is created by Scriberia with The Turing Way community, used under a CC-BY 4.0 licence. DOI: 10.5281/zenodo.3332807

[2] (Baker 2016)

Reproducibility crisis

- Reminder: 4 definitions of reproducible research[1]
- Our focus will be on REPRODUCIBILITY
- Most scientists report that they have failed to reproduce an experiment[2]
- But what about modelling?



[1] The Turing Way Community. This illustration is created by Scriberia with The Turing Way community, used under a CC-BY 4.0 licence. DOI: 10.5281/zenodo.3332807

[2] (Baker 2016)

Reproducibility crisis in modelling studies?

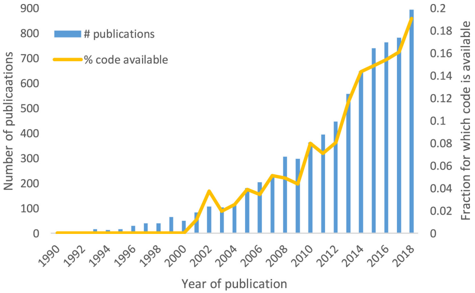
- *How often do you manage to find and run the code for a paper without issues?*

Reproducibility crisis in modelling studies?

- *How often do you manage to find and run the code for a paper without issues?*
- Not many studies have investigated reproducibility in modelling papers...

Reproducibility crisis in modelling studies?

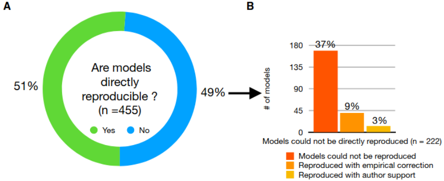
- *How often do you manage to find and run the code for a paper without issues?*
- Not many studies have investigated reproducibility in modelling papers...
  - Out of 7500 papers about individual-based and agent-based models only 11.2% provided their code[1]



[1] (Janssen, Pritchard, and Lee 2020)

Reproducibility crisis in modelling studies?

- *How often do you manage to find and run the code for a paper without issues?*
- Not many studies have investigated reproducibility in modelling papers...
    - Out of 7500 papers about individual-based and agent-based models only 11.2% provided their code[1]
    - Out of 455 ODE models from www.biomodels.org 49% are not directly reproducible[2]



[1] (Janssen, Pritchard, and Lee 2020)

[2] (Tiwari et al. 2021)

Reproducibility crisis in modelling studies!

Main barriers to reproducibility[1]:

- **"Dependency Hell"**: it might be difficult to recreate the original computational environment

[1] (Boettiger 2015)

Reproducibility crisis in modelling studies!

Main barriers to reproducibility[1]:

- **"Dependency Hell"**: it might be difficult to recreate the original computational environment
- **Imprecise documentation**: hard to install and build the code, without all parameter values impossible to recreate the results

[1] (Boettiger 2015)

Reproducibility crisis in modelling studies!

Main barriers to reproducibility[1]:

- **"Dependency Hell"**: it might be difficult to recreate the original computational environment
- **Imprecise documentation**: hard to install and build the code, without all parameter values impossible to recreate the results
- **Code rot**: software dependencies get updated all the time, so without proper maintenance the code will become unrunable at some point

[1] (Boettiger 2015)

Reproducibility crisis in modelling studies!

Main barriers to reproducibility[1]:

- **"Dependency Hell"**: it might be difficult to recreate the original computational environment
- **Imprecise documentation**: hard to install and build the code, without all parameter values impossible to recreate the results
- **Code rot**: software dependencies get updated all the time, so without proper maintenance the code will become unrunable at some point

What could help us solve these problems?[1]

[1] (Boettiger 2015)

Reproducibility crisis in modelling studies!

Main barriers to reproducibility[1]:

- **"Dependency Hell"**: it might be difficult to recreate the original computational environment
- **Imprecise documentation**: hard to install and build the code, without all parameter values impossible to recreate the results
- **Code rot**: software dependencies get updated all the time, so without proper maintenance the code will become unrunable at some point

What could help us solve these problems?[1]

- **Virtual machines**
  - Computationally heavy and not scalable (hard to combine code from multiple studies)
  - "Black box" without clear list of dependencies

[1] (Boettiger 2015)

Reproducibility crisis in modelling studies!

Main barriers to reproducibility[1]:

- **"Dependency Hell"**: it might be difficult to recreate the original computational environment
- **Imprecise documentation**: hard to install and build the code, without all parameter values impossible to recreate the results
- **Code rot**: software dependencies get updated all the time, so without proper maintenance the code will become unrunable at some point

What could help us solve these problems?[1]

- **Virtual machines**
  - Computationally heavy and not scalable (hard to combine code from multiple studies)
  - "Black box" without clear list of dependencies
- **Workflows**
  - A lot of proprietary formats
  - Limited functionality

[1] (Boettiger 2015)

Reproducibility crisis in modelling studies!

Main barriers to reproducibility[1]:

- **"Dependency Hell"**: it might be difficult to recreate the original computational environment
- **Imprecise documentation**: hard to install and build the code, without all parameter values impossible to recreate the results
- **Code rot**: software dependencies get updated all the time, so without proper maintenance the code will become unrunable at some point

What could help us solve these problems?[1]

- **Virtual machines**
  - Computationally heavy and not scalable (hard to combine code from multiple studies)
  - "Black box" without clear list of dependencies
- **Workflows**
  - A lot of proprietary formats
  - Limited functionality
- **Docker**
  - Lightweight and easily scalable
  - Clear list of dependencies
  - Open source
  - Linux functionality

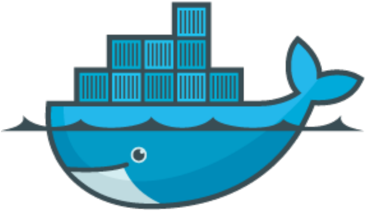
[1] (Boettiger 2015)

Learning objectives

- Docker becomes the new standard for reproducibility in science (e.g. Methods in Ecology and Evolution require the code to be Dockerized).

Learning objectives

- Docker becomes the new standard for reproducibility in science (e.g. Methods in Ecology and Evolution require the code to be Dockerized).
- What should ideally be achieved by the end of the talk:
  - **Understanding the basics of Docker and its components**
  - Being able to to create and manage Docker containers

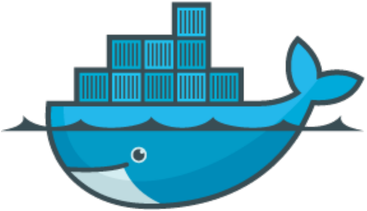


Learning objectives

- Docker becomes the new standard for reproducibility in science (e.g. Methods in Ecology and Evolution require the code to be Dockerized).
- What should ideally be achieved by the end of the talk:
  - **Understanding the basics of Docker and its components**
  - Being able to to create and manage Docker containers
- What is not going to be covered:
  - Docker automations for R (see Chan and Schoch 2023)
  - Docker desktop app

Learning objectives

- Docker becomes the new standard for reproducibility in science (e.g. Methods in Ecology and Evolution require the code to be Dockerized).
- What should ideally be achieved by the end of the talk:
  - **Understanding the basics of Docker and its components**
  - Being able to to create and manage Docker containers
- What is not going to be covered:
  - Docker automations for R (see Chan and Schoch 2023)
  - Docker desktop app



These slides are baked with Rmd and Dockerized! Go to github.com/enne-anastasia/docker-for-reproducible-research and see how I used Docker!

What is Docker?

Main idea

- Alice makes a code supplementary that Bob wants to reproduce.

What is Docker?

Main idea

- Alice makes a code supplementary that Bob wants to reproduce.
- With her supplementary Alice provides a small text file, that is both human-readable and machine-readable, and documents all the dependencies, as well as steps one needs to take in order to reproduce her analysis.

What is Docker?

Main idea

- Alice makes a code supplementary that Bob wants to reproduce.
- With her supplementary Alice provides a small text file, that is both human-readable and machine-readable, and documents all the dependencies, as well as steps one needs to take in order to reproduce her analysis.
- Bob can use Docker software to build an exact copy of Alice's environment from this text file, that will act like a virtual machine, but use much less resources.

What is Docker?

Main idea

- Alice makes a code supplementary that Bob wants to reproduce.
- With her supplementary Alice provides a small text file, that is both human-readable and machine-readable, and documents all the dependencies, as well as steps one needs to take in order to reproduce her analysis.
- Bob can use Docker software to build an exact copy of Alice's environment from this text file, that will act like a virtual machine, but use much less resources.
- If Bob doesn't want to be bothered with using Docker he still can read the file and install all the dependencies on his own device manually.

What is Docker?

Main idea

- Alice makes a code supplementary that Bob wants to reproduce.
- With her supplementary Alice provides a small text file, that is both human-readable and machine-readable, and documents all the dependencies, as well as steps one needs to take in order to reproduce her analysis.
- Bob can use Docker software to build an exact copy of Alice's environment from this text file, that will act like a virtual machine, but use much less resources.
- If Bob doesn't want to be bothered with using Docker he still can read the file and install all the dependencies on his own device manually.
- Alice is certain that she provided all the dependencies, because she tested this file with the Docker software.

What is Docker?

`docker build` ⇒ `docker run` ⇒

Docker file

- Text file with "source code" of the image:
  - Instructions on how to build the image
  - Commands to run your project

Docker image

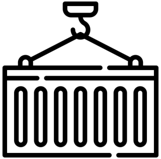
- An executable snapshot of a container
- Includes all dependencies needed to run a container

Docker container

- A running instance of the image
- Your "pocket Ubuntu"
- Self-contained

What is Docker?

More about Docker containers

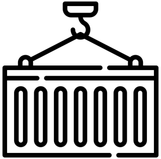


Docker container

- Containers stop existing after you exit them, meaning that all changes within the containers are lost.
- Containers are isolated from your file systems.

What is Docker?

More about Docker containers



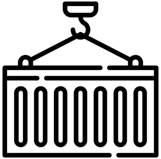
Docker container

- Containers stop existing after you exit them, meaning that all changes within the containers are lost.
- Containers are isolated from your file systems.

- Consequently, you need to either
  - copy your files into the image (with a command in your Docker file), or
  - mount a folder to a container (during `docker run`)

What is Docker?

More about Docker containers



Docker container

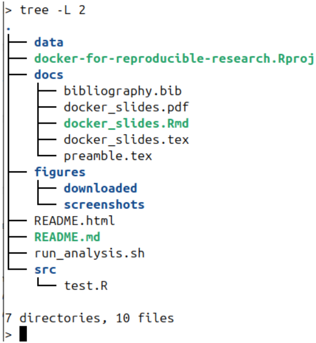
- Containers stop existing after you exit them, meaning that all changes within the containers are lost.
- Containers are isolated from your file systems.

- Consequently, you need to either
  - copy your files into the image (with a command in your Docker file), or
  - mount a folder to a container (during `docker run`)
- The rule of thumb is:
  - We MOUNT everything that takes plenty of space: e.g., data
  - We MOUNT everything that we want to be changed: e.g., figures
  - We COPY source code

Dockerizing an existing project

What project are we going to Dockerize?

- We will Dockerize the R project that I used to prepare this presentation
- It has a very common structure for a code supplementary:
    - data — folder containing all the data used in this project (empty in this case)
    - docs — folder with Rmd file that generates these slides
    - figures — folder with all the figures I used
    - README file
    - run_analysis.sh — one script to execute this project
    - src — source code (1 test R script in this case)



```
> tree -L 2
.
├── data
├── docker-for-reproducible-research.Rproj
├── docs
│   ├── bibliography.bib
│   ├── docker_slides.pdf
│   ├── docker_slides.Rmd
│   ├── docker_slides.tex
│   └── preamble.tex
├── figures
│   ├── downloaded
│   └── screenshots
├── README.html
├── README.md
├── run_analysis.sh
└── src
    └── test.R

7 directories, 10 files
> 
```

Dockerizing an existing project

STEP 1: Installing Docker

- The official docs.docker.com provides full manuals on how to install Docker:
    - To work from the command line on Linux install Docker Engine (**this is what I did** on Ubuntu)

Dockerizing an existing project

STEP 1: Installing Docker

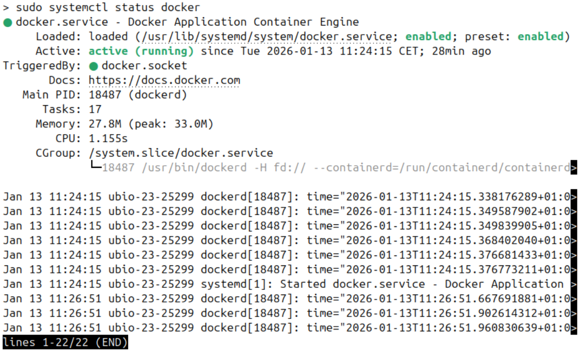
- The official docs.docker.com provides full manuals on how to install Docker:
  - To work from the command line on Linux install Docker Engine (**this is what I did** on Ubuntu)
  - To work with UI install Docker Desktop for Linux, Windows, or Mac (beware, this is less "light-weight" compared to Docker in command line)

Dockerizing an existing project

STEP 1: Installing Docker

- The official docs.docker.com provides full manuals on how to install Docker:
    - To work from the command line on Linux install Docker Engine (**this is what I did** on Ubuntu)
    - To work with UI install Docker Desktop for Linux, Windows, or Mac (beware, this is less "light-weight" compared to Docker in command line)
- Verify that the installation is successful:

  ```
  sudo systemctl status docker
  sudo docker run hello-world
  ```



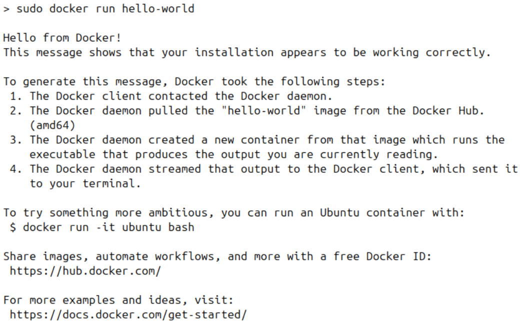
```
> sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: active (running) since Tue 2026-01-13 11:24:15 CET; 28min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 18487 (dockerd)
      Tasks: 17
     Memory: 27.8M (peak: 33.0M)
        CPU: 1.155s
     CGroup: /system.slice/docker.service
             └─18487 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd>

Jan 13 11:24:15 ubio-23-25299 dockerd[18487]: time="2026-01-13T11:24:15.338176289+01:0>
Jan 13 11:24:15 ubio-23-25299 dockerd[18487]: time="2026-01-13T11:24:15.349587902+01:0>
Jan 13 11:24:15 ubio-23-25299 dockerd[18487]: time="2026-01-13T11:24:15.349839905+01:0>
Jan 13 11:24:15 ubio-23-25299 dockerd[18487]: time="2026-01-13T11:24:15.368402040+01:0>
Jan 13 11:24:15 ubio-23-25299 dockerd[18487]: time="2026-01-13T11:24:15.376681433+01:0>
Jan 13 11:24:15 ubio-23-25299 dockerd[18487]: time="2026-01-13T11:24:15.37677321l+01:0>
Jan 13 11:24:15 ubio-23-25299 systemd[1]: Started docker.service - Docker Application >
Jan 13 11:26:51 ubio-23-25299 dockerd[18487]: time="2026-01-13T11:26:51.667691881+01:0>
Jan 13 11:26:51 ubio-23-25299 dockerd[18487]: time="2026-01-13T11:26:51.902614312+01:0>
Jan 13 11:26:51 ubio-23-25299 dockerd[18487]: time="2026-01-13T11:26:51.960830639+01:0>
lines 1-22/22 (END)
```

Dockerizing an existing project

STEP 1: Installing Docker

- The official docs.docker.com provides full manuals on how to install Docker:
  - To work from the command line on Linux install Docker Engine (**this is what I did** on Ubuntu)
  - To work with UI install Docker Desktop for Linux, Windows, or Mac (beware, this is less "light-weight" compared to Docker in command line)
- Verify that the installation is successful:

```
sudo systemctl status docker
sudo docker run hello-world
```



```
> sudo docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Dockerizing an existing project

STEP 2: Trying the basic Docker commands

- **Starting Docker:**

  ```
  sudo systemctl start docker
  ```

  (Docker is running in the background waiting for you to use it)

- **Stopping Docker:**

  ```
  sudo systemctl stop docker
  ```

  (Docker is deactivated, but sometimes `docker.socket` remains active)

- **Checking Docker status:**

  ```
  sudo systemctl status docker
  ```

Dockerizing an existing project

STEP 2: Trying the basic Docker commands

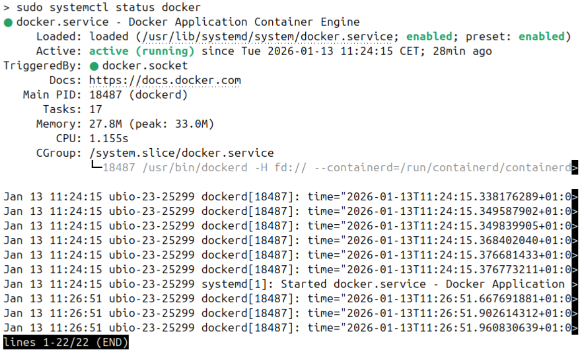
- **Starting Docker:**

  `sudo systemctl start docker`

  (Docker is running in the background waiting for you to use it)

- **Checking Docker status:**

  `sudo systemctl status docker`

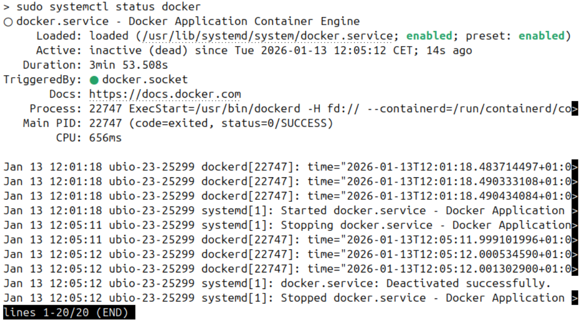


```
> sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: active (running) since Tue 2026-01-13 11:24:15 CET; 28min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 18487 (dockerd)
      Tasks: 17
     Memory: 27.8M (peak: 33.0M)
        CPU: 1.155s
     CGroup: /system.slice/docker.service
             └─18487 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd>

Jan 13 11:24:15 ubio-23-25299 dockerd[18487]: time="2026-01-13T11:24:15.338176289+01:0>
Jan 13 11:24:15 ubio-23-25299 dockerd[18487]: time="2026-01-13T11:24:15.349587902+01:0>
Jan 13 11:24:15 ubio-23-25299 dockerd[18487]: time="2026-01-13T11:24:15.349839905+01:0>
Jan 13 11:24:15 ubio-23-25299 dockerd[18487]: time="2026-01-13T11:24:15.368402040+01:0>
Jan 13 11:24:15 ubio-23-25299 dockerd[18487]: time="2026-01-13T11:24:15.376681433+01:0>
Jan 13 11:24:15 ubio-23-25299 dockerd[18487]: time="2026-01-13T11:24:15.376773211+01:0>
Jan 13 11:24:15 ubio-23-25299 systemd[1]: Started docker.service - Docker Application >
Jan 13 11:26:51 ubio-23-25299 dockerd[18487]: time="2026-01-13T11:26:51.667691881+01:0>
Jan 13 11:26:51 ubio-23-25299 dockerd[18487]: time="2026-01-13T11:26:51.902614312+01:0>
Jan 13 11:26:51 ubio-23-25299 dockerd[18487]: time="2026-01-13T11:26:51.960830639+01:0>
lines 1-22/22 (END)
```

Dockerizing an existing project

STEP 2: Trying the basic Docker commands

- **Stopping Docker:**
  `sudo systemctl stop docker`
  (Docker is deactivated, but sometimes `docker.socket` remains active)
- **Checking Docker status:**
  `sudo systemctl status docker`

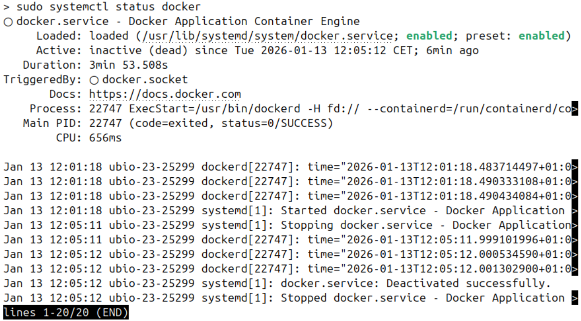


```
> sudo systemctl status docker
○ docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: inactive (dead) since Tue 2026-01-13 12:05:12 CET; 14s ago
   Duration: 3min 53.508s
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
    Process: 22747 ExecStart=/usr/bin/dockerd -H fd:// --containerd=/run/containerd/co>
   Main PID: 22747 (code=exited, status=0/SUCCESS)
        CPU: 656ms

Jan 13 12:01:18 ubio-23-25299 dockerd[22747]: time="2026-01-13T12:01:18.483714497+01:0>
Jan 13 12:01:18 ubio-23-25299 dockerd[22747]: time="2026-01-13T12:01:18.490333108+01:0>
Jan 13 12:01:18 ubio-23-25299 dockerd[22747]: time="2026-01-13T12:01:18.490434084+01:0>
Jan 13 12:01:18 ubio-23-25299 systemd[1]: Started docker.service - Docker Application >
Jan 13 12:05:11 ubio-23-25299 systemd[1]: Stopping docker.service - Docker Application>
Jan 13 12:05:11 ubio-23-25299 dockerd[22747]: time="2026-01-13T12:05:11.999101996+01:0>
Jan 13 12:05:12 ubio-23-25299 dockerd[22747]: time="2026-01-13T12:05:12.000534590+01:0>
Jan 13 12:05:12 ubio-23-25299 dockerd[22747]: time="2026-01-13T12:05:12.001302900+01:0>
Jan 13 12:05:12 ubio-23-25299 systemd[1]: docker.service: Deactivated successfully.
Jan 13 12:05:12 ubio-23-25299 systemd[1]: Stopped docker.service - Docker Application >
lines 1-20/20 (END)
```

Dockerizing an existing project

STEP 2: Trying the basic Docker commands

- **Stopping Docker:**
  `sudo systemctl stop docker`
  (Docker is deactivated, but sometimes `docker.socket` remains active)
- **Checking Docker status:**
  `sudo systemctl status docker`
- To kill `docker.socket` use
  `sudo systemctl stop docker.socket`



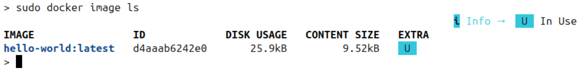
```
> sudo systemctl status docker
○ docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: inactive (dead) since Tue 2026-01-13 12:05:12 CET; 6min ago
   Duration: 3min 53.508s
TriggeredBy: ○ docker.socket
       Docs: https://docs.docker.com
    Process: 22747 ExecStart=/usr/bin/dockerd -H fd:// --containerd=/run/containerd/co>
   Main PID: 22747 (code=exited, status=0/SUCCESS)
        CPU: 656ms

Jan 13 12:01:18 ubio-23-25299 dockerd[22747]: time="2026-01-13T12:01:18.483714497+01:0>
Jan 13 12:01:18 ubio-23-25299 dockerd[22747]: time="2026-01-13T12:01:18.490333108+01:0>
Jan 13 12:01:18 ubio-23-25299 dockerd[22747]: time="2026-01-13T12:01:18.490434084+01:0>
Jan 13 12:01:18 ubio-23-25299 systemd[1]: Started docker.service - Docker Application >
Jan 13 12:05:11 ubio-23-25299 systemd[1]: Stopping docker.service - Docker Application>
Jan 13 12:05:11 ubio-23-25299 dockerd[22747]: time="2026-01-13T12:05:11.999101996+01:0>
Jan 13 12:05:12 ubio-23-25299 dockerd[22747]: time="2026-01-13T12:05:12.000534590+01:0>
Jan 13 12:05:12 ubio-23-25299 dockerd[22747]: time="2026-01-13T12:05:12.001302900+01:0>
Jan 13 12:05:12 ubio-23-25299 systemd[1]: docker.service: Deactivated successfully.
Jan 13 12:05:12 ubio-23-25299 systemd[1]: Stopped docker.service - Docker Application >
lines 1-20/20 (END)
```

Dockerizing an existing project

STEP 2: Trying the basic Docker commands

- **Listing all available Docker images:**

  `sudo docker image ls`

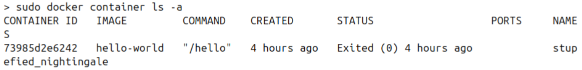


```
> sudo docker image ls
                                                                 i Info → U In Use
IMAGE                ID             DISK USAGE   CONTENT SIZE   EXTRA
hello-world:latest   d4aaab6242e0       25.9kB         9.52kB   U
> 
```

Dockerizing an existing project

STEP 2: Trying the basic Docker commands

- **Listing all available Docker images:**
  `sudo docker image ls`
- **Listing all containers that are currently running:**
  `sudo docker container ls -a`



```
> sudo docker container ls -a
CONTAINER ID   IMAGE         COMMAND    CREATED       STATUS                   PORTS     NAME
S
73985d2e6242   hello-world   "/hello"   4 hours ago   Exited (0) 4 hours ago             stup
efied_nightingale
```

Dockerizing an existing project

STEP 2: Trying the basic Docker commands

- **Listing all available Docker images:**

  `sudo docker image ls`

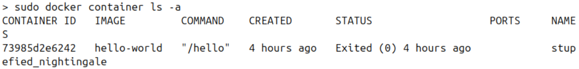
- **Listing all containers that are currently running:**

  `sudo docker container ls -a`

- **Deleting a container:**

  `sudo docker container rm <ID>`

  `sudo docker container rm <NAME>`



```
> sudo docker container ls -a
CONTAINER ID   IMAGE         COMMAND    CREATED         STATUS                     PORTS     NAME
S
73985d2e6242   hello-world   "/hello"   4 hours ago     Exited (0) 4 hours ago               stup
efied_nightingale
```

Dockerizing an existing project

STEP 2: Trying the basic Docker commands

- **Listing all available Docker images:**

  `sudo docker image ls`
- **Listing all containers that are currently running:**

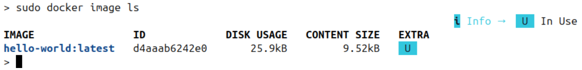
  `sudo docker container ls -a`
- **Deleting a container:**

  `sudo docker container rm <ID>`

  `sudo docker container rm <NAME>`
- **Deleting an image:**

  `sudo docker image rm <ID>`

  `sudo docker image rm <IMAGE>`

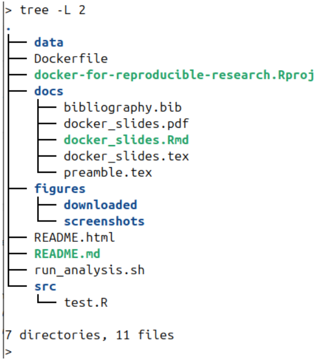


```
> sudo docker image ls
                                                                          i Info → U In Use
IMAGE                ID             DISK USAGE   CONTENT SIZE   EXTRA
hello-world:latest   d4aaab6242e0       25.9kB         9.52kB   U
> 
```

Dockerizing an existing project

STEP 3: Writing the Docker file

1. Create an empty `Dockerfile` in the root directory of you project

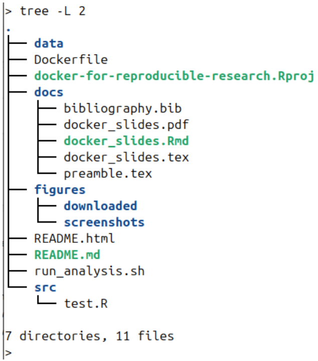


```
> tree -L 2
.
├── data
├── Dockerfile
├── docker-for-reproducible-research.Rproj
├── docs
│   ├── bibliography.bib
│   ├── docker_slides.pdf
│   ├── docker_slides.Rmd
│   ├── docker_slides.tex
│   └── preamble.tex
├── figures
│   ├── downloaded
│   └── screenshots
├── README.html
├── README.md
├── run_analysis.sh
└── src
    └── test.R

7 directories, 11 files
>
```

Dockerizing an existing project

STEP 3: Writing the Docker file

1. Create an empty `Dockerfile` in the root directory of you project
2. Pick the base image
   - Writing your Docker file from scratch is like setting a new work laptop from scratch
   - Thankfully, we do not have to do that because there are Docker images we can build upon!

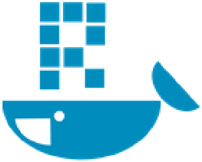


```
> tree -L 2
.
├── data
├── Dockerfile
├── docker-for-reproducible-research.Rproj
├── docs
│   ├── bibliography.bib
│   ├── docker_slides.pdf
│   ├── docker_slides.Rmd
│   ├── docker_slides.tex
│   └── preamble.tex
├── figures
│   ├── downloaded
│   └── screenshots
├── README.html
├── README.md
├── run_analysis.sh
└── src
    └── test.R

7 directories, 11 files
>
```

Dockerizing an existing project

STEP 3: Writing the Docker file

1. Create an empty `Dockerfile` in the root directory of you project
2. Pick the base image
   - Writing your Docker file from scratch is like setting a new work laptop from scratch
   - Thankfully, we do not have to do that because there are Docker images we can build upon!
   - For this project I will use `rocker` — Docker image with pre-installed R



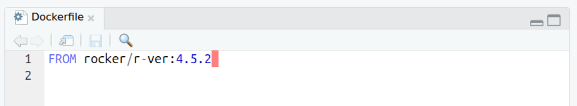
The Rocker Project

Docker Containers for the R Environment

Dockerizing an existing project

STEP 3: Writing the Docker file

3. Write the first line in your `Dockerfile`:
   - We build our image starting from Docker image with pre-installed R
   - I specify the version 4.5.2 of R since this is the version that I currently have on my laptop
   - There can be only one `FROM` image!



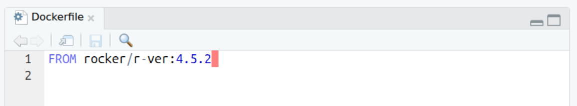
```
FROM rocker/r-ver:4.5.2
```

Dockerizing an existing project

STEP 3: Writing the Docker file

3. Write the first line in your `Dockerfile`:
   - We build our image starting from Docker image with pre-installed R
   - I specify the version 4.5.2 of R since this is the version that I currently have on my laptop
   - There can be only one `FROM` image!

- Congratulations! This is the the minimal Docker file we can use to build and run a Docker image (atlhough, not very usefull one)

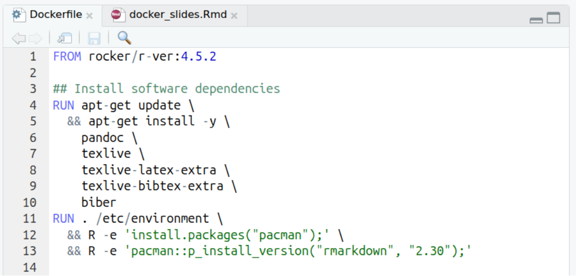


```
FROM rocker/r-ver:4.5.2
```

Dockerizing an existing project

STEP 3: Writing the Docker file

4. Specify software dependencies
   - It is okay to do that in iterations. When you think you got all the dependencies, go to the next step and see if you can run your project in the Docker container.
   - Here I have 2 types of dependencies: OS-level and R-level.

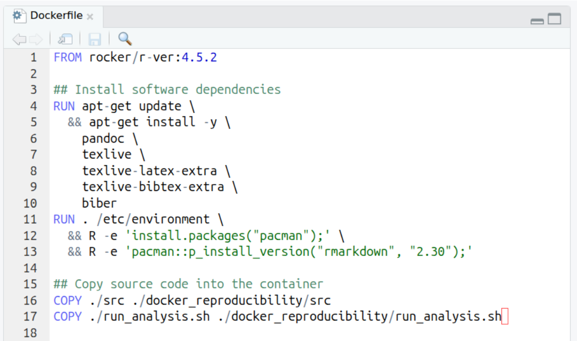


```
FROM rocker/r-ver:4.5.2

## Install software dependencies
RUN apt-get update \
  && apt-get install -y \
    pandoc \
    texlive \
    texlive-latex-extra \
    texlive-bibtex-extra \
    biber
RUN . /etc/environment \
  && R -e 'install.packages("pacman");' \
  && R -e 'pacman::p_install_version("rmarkdown", "2.30");'

```

Dockerizing an existing project

STEP 3: Writing the Docker file

5. Copy the source code into the image
   - The inside of the container is an Ubuntu file system with installed dependencies, therefore I copy everything into a separate folder to keep things neat.

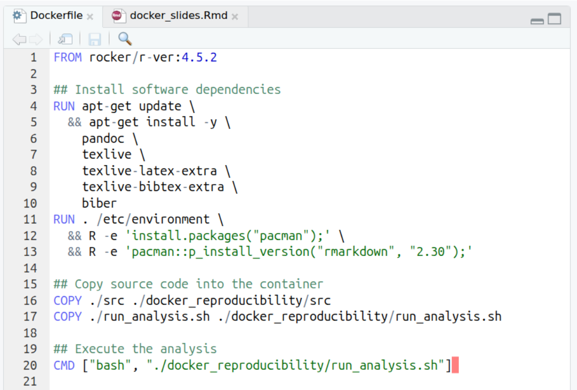


```
FROM rocker/r-ver:4.5.2

## Install software dependencies
RUN apt-get update \
  && apt-get install -y \
    pandoc \
    texlive \
    texlive-latex-extra \
    texlive-bibtex-extra \
    biber
RUN . /etc/environment \
  && R -e 'install.packages("pacman");' \
  && R -e 'pacman::p_install_version("rmarkdown", "2.30");'

## Copy source code into the container
COPY ./src ./docker_reproducibility/src
COPY ./run_analysis.sh ./docker_reproducibility/run_analysis.sh

```

Dockerizing an existing project

STEP 3: Writing the Docker file

5. Copy the source code into the image
   - The inside of the container is an Ubuntu file system with installed dependencies, therefore I copy everything into a separate folder to keep things neat.
6. Write the last line with the command to execute the analysis
   - There can be only one `CMD` line in your Docker file!

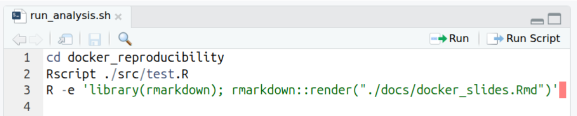


```
FROM rocker/r-ver:4.5.2

## Install software dependencies
RUN apt-get update \
  && apt-get install -y \
    pandoc \
    texlive \
    texlive-latex-extra \
    texlive-bibtex-extra \
    biber
RUN . /etc/environment \
  && R -e 'install.packages("pacman");' \
  && R -e 'pacman::p_install_version("rmarkdown", "2.30");'

## Copy source code into the container
COPY ./src ./docker_reproducibility/src
COPY ./run_analysis.sh ./docker_reproducibility/run_analysis.sh

## Execute the analysis
CMD ["bash", "./docker_reproducibility/run_analysis.sh"]
```

Dockerizing an existing project

STEP 3: Writing the Docker file

5. Copy the source code into the image
   - The inside of the container is an Ubuntu file system with installed dependencies, therefore I copy everything into a separate folder to keep things neat.
6. Write the last line with the command to execute the analysis
   - There can be only one CMD line in your Docker file!



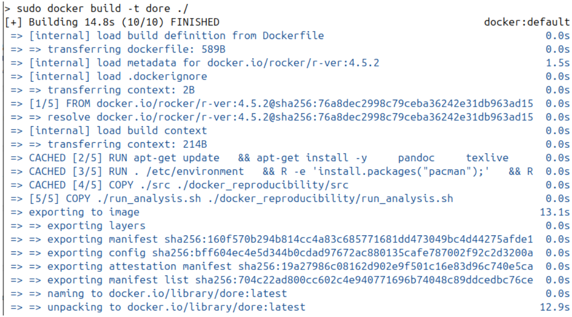
```
cd docker_reproducibility
Rscript ./src/test.R
R -e 'library(rmarkdown); rmarkdown::render("./docs/docker_slides.Rmd")'

```

Dockerizing an existing project

STEP 4: Building the image

```
sudo docker build -t <TAG_NAME> ./
```

- It probably will take a while to install all dependencies
- When you change one layer of your image and build again, Docker will execute only starting from layers that were changed



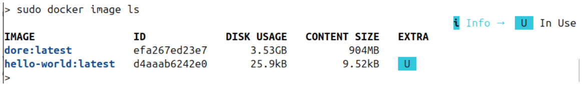
```
> sudo docker build -t dore ./
[+] Building 14.8s (10/10) FINISHED                                           docker:default
 => [internal] load build definition from Dockerfile                                    0.0s
 => => transferring dockerfile: 589B                                                    0.0s
 => [internal] load metadata for docker.io/rocker/r-ver:4.5.2                           1.5s
 => [internal] load .dockerignore                                                       0.0s
 => => transferring context: 2B                                                         0.0s
 => [1/5] FROM docker.io/rocker/r-ver:4.5.2@sha256:76a8dec2998c79ceba36242e31db963ad15  0.0s
 => => resolve docker.io/rocker/r-ver:4.5.2@sha256:76a8dec2998c79ceba36242e31db963ad15  0.0s
 => [internal] load build context                                                       0.0s
 => => transferring context: 214B                                                       0.0s
 => CACHED [2/5] RUN apt-get update   && apt-get install -y     pandoc     texlive      0.0s
 => CACHED [3/5] RUN . /etc/environment   && R -e 'install.packages("pacman");'   && R  0.0s
 => CACHED [4/5] COPY ./src ./docker_reproducibility/src                                0.0s
 => [5/5] COPY ./run_analysis.sh ./docker_reproducibility/run_analysis.sh               0.0s
 => exporting to image                                                                 13.1s
 => => exporting layers                                                                 0.0s
 => => exporting manifest sha256:160f570b294b814cc4a83c685771681dd473049bc4d44275afde1  0.0s
 => => exporting config sha256:bff604ec4e5d344b0cdad97672ac880135cafe787002f92c2d3200a  0.0s
 => => exporting attestation manifest sha256:19a27986c08162d902e9f501c16e83d96c740e5ca  0.0s
 => => exporting manifest list sha256:704c22ad800cc602c4e940771696b74048c89ddcedbc76ce  0.0s
 => => naming to docker.io/library/dore:latest                                          0.0s
 => => unpacking to docker.io/library/dore:latest                                      12.9s
```

Dockerizing an existing project

STEP 4: Building the image

```
sudo docker build -t <TAG_NAME> ./
```

- It probably will take a while to install all dependencies
- When you change one layer of your image and build again, Docker will execute only starting from layers that were changed



```
> sudo docker image ls
                                                                        i Info →  U  In Use
IMAGE                ID             DISK USAGE   CONTENT SIZE   EXTRA
dore:latest          efa267ed23e7       3.53GB          904MB
hello-world:latest   d4aaab6242e0       25.9kB         9.52kB   U
>
```

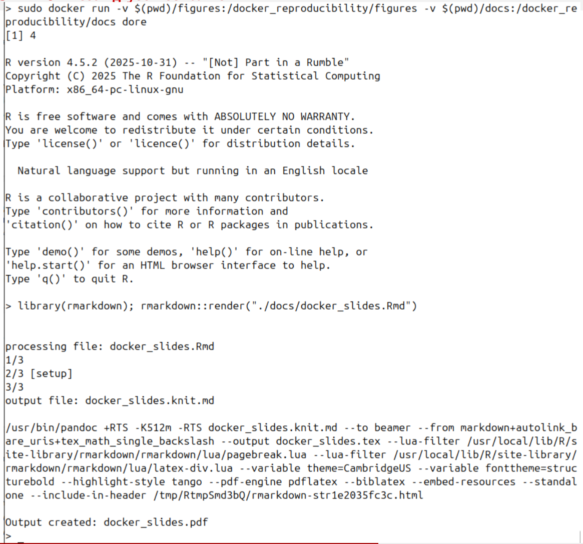
Dockerizing an existing project

STEP 5: Testing your container

```
sudo docker run -v $(pwd)/figures:/docker_reproducibility/figures -v
$(pwd)/docs:/docker_reproducibility/docs <TAG_NAME>
```

Dockerizing an existing project

STEP 5: Testing your container



```
> sudo docker run -v $(pwd)/figures:/docker_reproducibility/figures -v $(pwd)/docs:/docker_re
producibility/docs dore
[1] 4

R version 4.5.2 (2025-10-31) -- "[Not] Part in a Rumble"
Copyright (C) 2025 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library(rmarkdown); rmarkdown::render("./docs/docker_slides.Rmd")


processing file: docker_slides.Rmd
1/3
2/3 [setup]
3/3
output file: docker_slides.knit.md

/usr/bin/pandoc +RTS -K512m -RTS docker_slides.knit.md --to beamer --from markdown+autolink_b
are_uris+tex_math_single_backslash --output docker_slides.tex --lua-filter /usr/local/lib/R/s
ite-library/rmarkdown/rmarkdown/lua/pagebreak.lua --lua-filter /usr/local/lib/R/site-library/
rmarkdown/rmarkdown/lua/latex-div.lua --variable theme=CambridgeUS --variable fonttheme=struc
turebold --highlight-style tango --pdf-engine pdflatex --biblatex --embed-resources --standal
one --include-in-header /tmp/RtmpSmd3bQ/rmarkdown-str1e2035fc3c.html

Output created: docker_slides.pdf
> _
```

- Baker, Monya (2016). “1,500 scientists lift the lid on reproducibility”. In: *Nature* 533.7604, pp. 452–454. DOI: 10.1038/533452a.
- Boettiger, Carl (2015). “An introduction to Docker for reproducible research”. In: *SIGOPS Oper. Syst. Rev.* 49.1, pp. 71–79. DOI: 10.1145/2723872.2723882.
- Chan, Chung-hong and David Schoch (2023). “rang: Reconstructing reproducible R computational environments”. In: *PLOS ONE* 18.6, e0286761. DOI: 10.1371/journal.pone.0286761.
- Janssen, Marco A., Calvin Pritchard, and Allen Lee (2020). “On code sharing and model documentation of published individual and agent-based models”. In: *Environmental Modelling & Software* 134, p. 104873. DOI: 10.1016/j.envsoft.2020.104873.
- Tiwari, Krishna et al. (2021). “Reproducibility in systems biology modelling”. In: *Molecular Systems Biology* 17.2, e9982. DOI: 10.15252/msb.20209982.