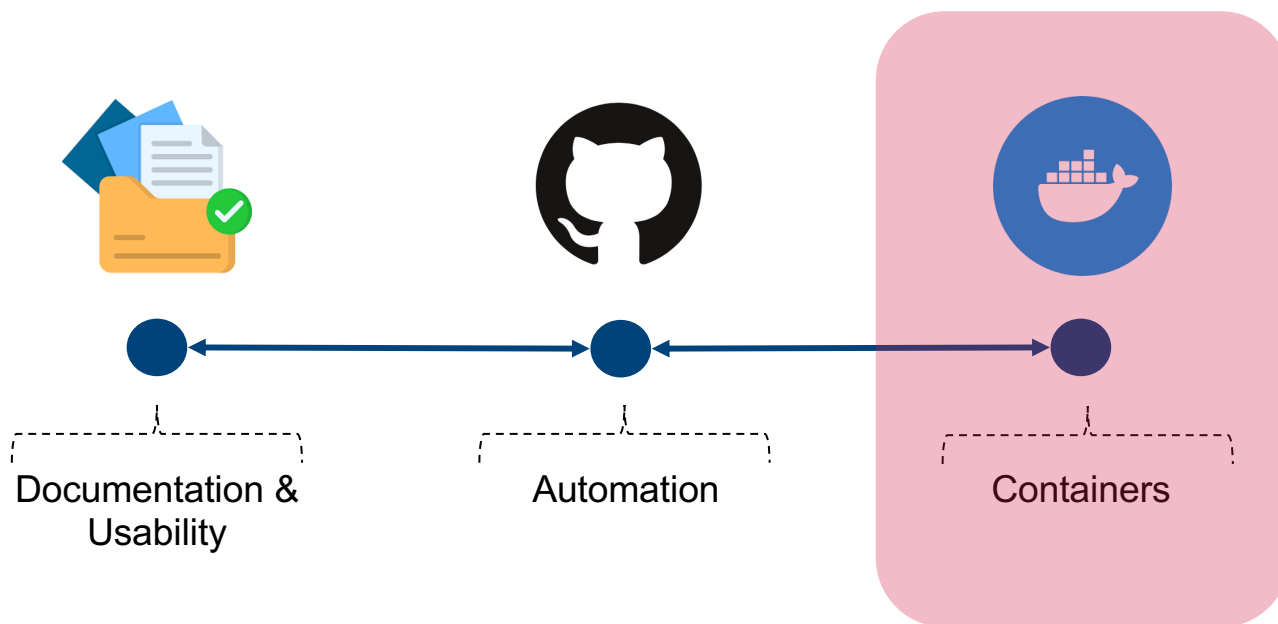


## Best practices for reproducibility 🎉



## Managing Packages and Environments

Your tool's software dependencies

Your new  
users



Photo from Rutpratheep through Pexels. Arrangement by Candace Savonen



# Documenting package versions in R

- Record a list of your packages
- `sessionInfo()`
- Even if you have the same packages installed between two computers, you can still get different results. This is because package versions do influence results as demonstrated by [Beaulieu-Jones and Greene, 2017](#).

1 Information about this notebook
2 Set up
3 Directories and paths to file Inputs/Outputs
4 Read seurat object
5 Number of cells per cluster before removing cell contamination
6 Remove cell contamination
7 Number of cells per cluster after removing cell contamination
8 Plot clusters after removing cell contamination
9 Process seurat object
10 Save output files
11 Session Info

## 11 Session Info

```
## R version 4.4.0 (2024-04-24)
## Platform: x86_64-pc-linux-gnu
## Running under: Red Hat Enterprise Linux 8.4 (Ootpa)
##
## Matrix products: default
## BLAS: /research/rgs01/applications/hpcf/authorized_apps/rhel8_apps/lapack/3.10.1/install/lib64/libblas.so.3.10.1
## LAPACK: /research/rgs01/applications/hpcf/authorized_apps/rhel8_apps/lapack/3.10.1/install/lib64/liblapack.so.3.10.1
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8      LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8     LC_NAME=C
##  [9] LC_ADDRESS=C             LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## time zone: America/Chicago
## tzcode source: system (glibc)
##
## attached base packages:
## [1] grid      stats      graphics  grDevices  utils      datasets  methods
## [8] base
##
## other attached packages:
## [1] RColorBrewer_1.1-3 patchwork_1.2.0 ggthemes_5.1.0 scooter_0.0.0.9004
## [5] SeuratObject_5.0.2 Seurat_4.4.0 future_1.33.2 devtools_2.4.5
## [9] usethis_2.2.3 lubridate_1.9.3 forcats_1.0.0 stringr_1.5.1
## [13] dplyr_1.1.4 purrr_1.0.2 readr_2.1.5 tidyr_1.3.1
## [17] tibble_3.2.1 ggplot2_3.5.1 tidyverse_2.0.0 yaml_2.3.8
##
## loaded via a namespace (and not attached):
## [1] jsonlite_1.8.8 magrittr_2.0.3 spatstat.utils_3.0-4
## [4] farver_2.1.2 rmarkdown_2.26 ragg_1.3.1
## [7] fs_1.6.4 vctrs_0.6.5 ROCR_1.0-11
## [10] memoise_2.0.1 spatstat.explore_3.2-7 htmltools_0.5.8.1
## [13] sass_0.4.9 sctransform_0.4.1 parallelly_1.37.1
## [16] KernSmooth_2.23-24 bslib_0.7.0 htmlwidgets_1.6.4
## [19] ica_1.0-3 plyr_1.8.9 plotly_4.10.4
## [22] zoo_1.8-12 cachem_1.0.8 igraph_2.0.3
```



## Package managers - share a useable snapshot of your environment

- R
  - renv is pretty useful, but it only gets us so far... only R packages (and a bit of Python, in some situations).
  - For software outside R, other package management systems are required.
- python
  - conda is one of the more popular and flexible package managers. Like renv, you can create separate sets of software with different versions for different projects.
  - Started as a Python package manager, but it can be used for any command line software.
  - [bioconda](#)



# Containers

- Docker and Singularity
  - “Containers” that include everything from the operating system up
  - Run one OS inside another, with all the things frozen to particular versions
  - Cloud platforms love containers.

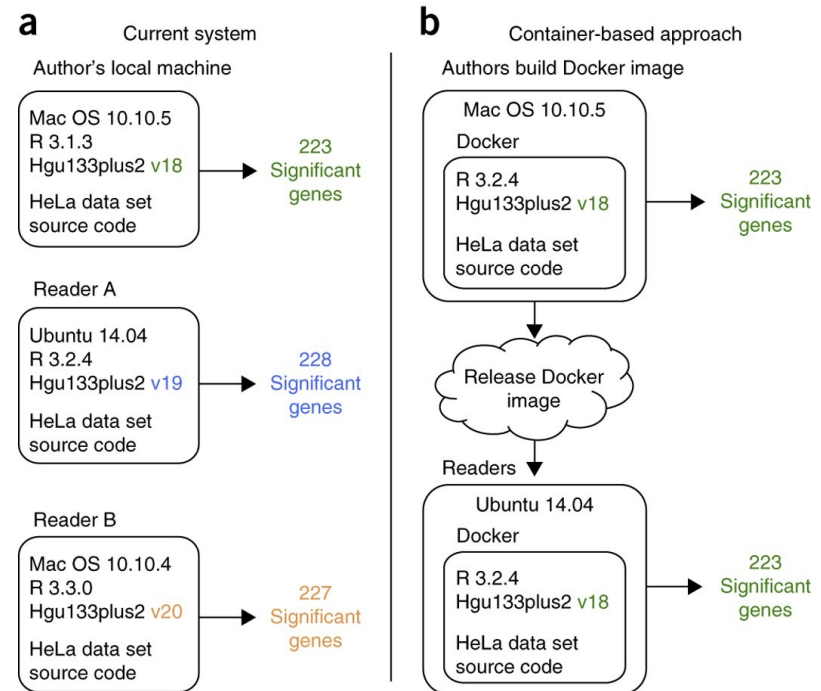


Figure from [Beaulieu-Jones and Greene, 2017](#)





## What is Docker?

- The same computing environment
- We can create a Docker image that any collaborator can use to run the analysis.
- This way we ensure that everyone is using the same computing environment.
- If there are any differences in results, it won't be because of version differences.
- “Docker allows you to package an application with all its dependencies into a standardized unit for software development”.

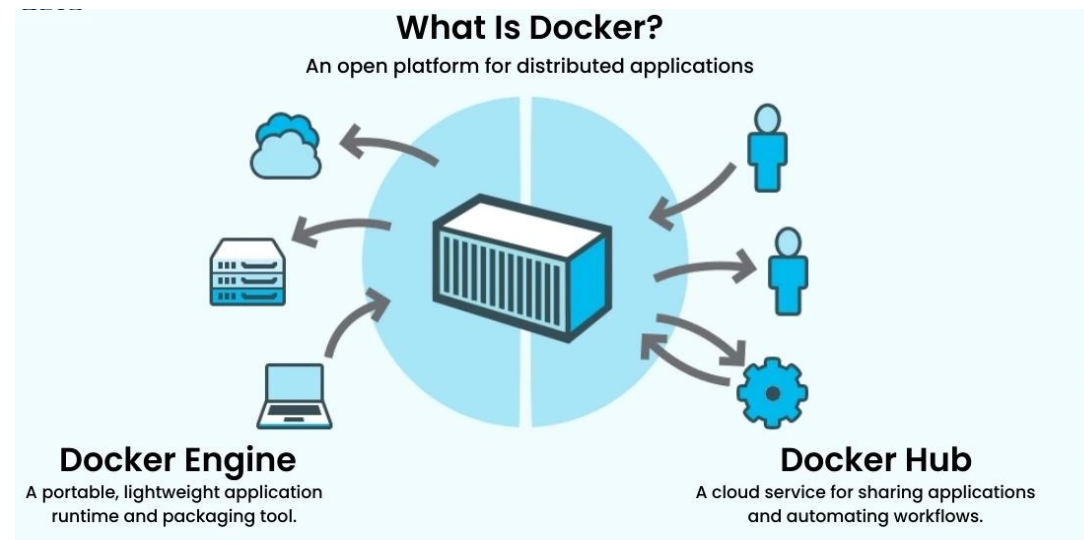
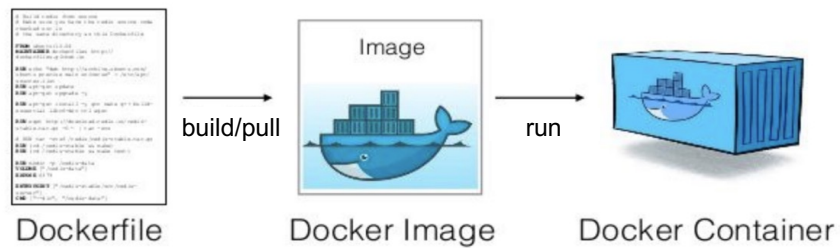


Figure from [mardsdevs](https://mardsdevs.com)





## Docker terminology



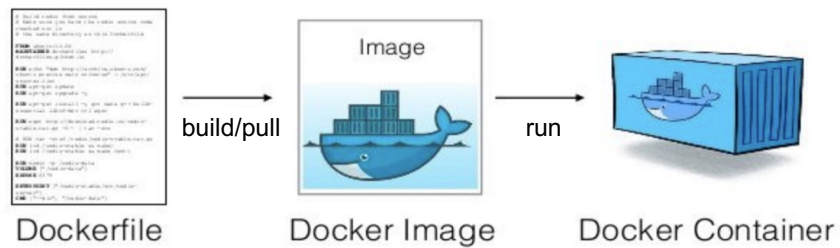
<b>Dockerfile</b>	yaml-based file that's used to build your image; this is what we can version control
-------------------	--

Figure is from: <https://cto.ai/blog/docker-image-vs-container-vs-dockerfile/>





## Docker terminology



<b>Dockerfile</b>	yaml-based file that's used to build your image; this is what we can version control
<b>Docker image</b>	snapshot of your computing environment that you can move from place to place. We can download images from online and then use them to make a container

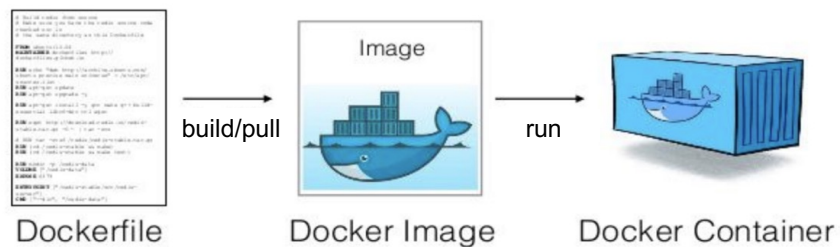
Figure is from: <https://cto.ai/blog/docker-image-vs-container-vs-dockerfile/>







## Docker terminology



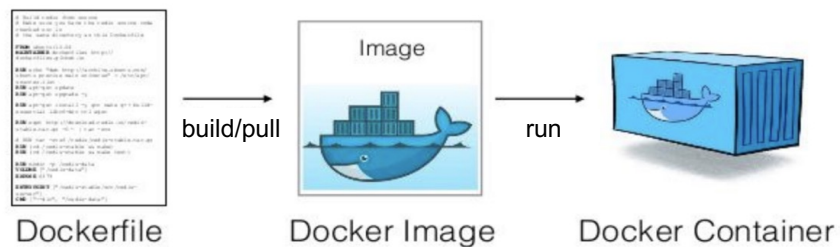
<b>Dockerfile</b>	yaml-based file that's used to build your image; this is what we can version control
<b>Docker image</b>	snapshot of your computing environment that you can move from place to place. We can download images from online and then use them to make a container
<b>Containers</b>	very small user-level virtualization that helps you build, install, and run your code and what we use to actually run analyses

Figure is from: <https://cto.ai/blog/docker-image-vs-container-vs-dockerfile/>





## Docker terminology



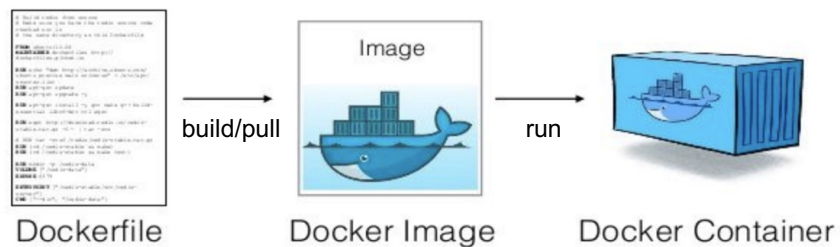
<b>Dockerfile</b>	yaml-based file that's used to build your image; this is what we can version control
<b>Docker image</b>	snapshot of your computing environment that you can move from place to place. We can download images from online and then use them to make a container
<b>Containers</b>	very small user-level virtualization that helps you build, install, and run your code and what we use to actually run analyses
<b>Dockerhub</b>	GitHub for your Docker images; you can set up Dockerhub to automatically build an image anytime you update your Dockerfile in GitHub

Figure is from: <https://cto.ai/blog/docker-image-vs-container-vs-dockerfile/>





## Docker terminology



<b>Dockerfile</b>	yaml-based file that's used to build your image; this is what we can version control
<b>Docker image</b>	snapshot of your computing environment that you can move from place to place. We can download images from online and then use them to make a container
<b>Containers</b>	very small user-level virtualization that helps you build, install, and run your code and what we use to actually run analyses
<b>Dockerhub</b>	GitHub for your Docker images; you can set up Dockerhub to automatically build an image anytime you update your Dockerfile in GitHub
<b>Registry</b>	A centralized service for storing, managing and distributing container images

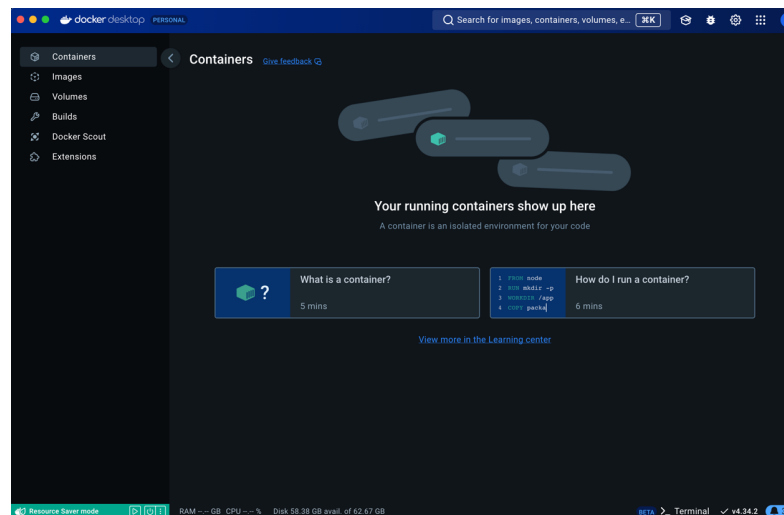
Figure is from: <https://cto.ai/blog/docker-image-vs-container-vs-dockerfile/>





## Install Docker manually

- How to [install Docker](#), following the instructions for your OS.
- For more details, see [./resources/05-install-docker-manually.md](#)





## How to use Docker on your local workstation

---

- For more details, see [./resources/06-how-to-use-docker-command-line.md](/resources/06-how-to-use-docker-command-line.md)
- This is a detailed guide on how to login, build custom/push/pull Docker images from your local workstation.
- [Managing Docker images](#)





## But wait... what if we work on HPC?

---

- Docker is **NOT** Root Safe!
- Docker grants root level permissions by default
- Fine for Kubernetes, but exposes security vulnerabilities on shared file systems.





## Using Containers on HPC

- **Singularity** (Apptainer)
  - Built with HPC systems in mind, to solve problems with Docker
    1. Security:
      - A user has the same permissions inside and outside the container
      - No root escalation is possible
    2. Local filesystem paths are easily accessed
    3. Simplifies the use of Multinode MPI and GPU's
    4. Integrated with resource managers such as Slurm
    5. Automatically converts Docker containers to Singularity or OCI format

Source: 'St. Jude Containerization in HPC Environments' workshop





## How to use Docker on HPC

- A detailed guide on how to pull Docker images from and convert them to sif files on HPC and run your module of preference
  - Dockerhub registry: [./resources/07-how-to-run-docker-image-from-HPC-with-Dockerhub-registry.md](#)
  - HPC registry: [./resources/08-how-to-run-docker-image-from-HPC-with-HPC-registry.md](#)
- To make our repos accessible to users outside of St Jude/HPC, we will be using Dockerhub registry for our projects. Except for cases that we are handling private repositories or are concerned for data breach.







## Our first PR is about testing and running Docker image 😊

- Create and build Dockerfile for the `single-cell-rna-analysis` repo → [PR#3](#)





## More resources 🎉

---

- [Docker home page](#)
- [Docker for beginners](#)
- [Docker for Data Scientists](#)
- [Nüst \*et al.\*, 2020](#)
- [Containers and HPC](#)
- [Singularity-tutorial.github.io](#)



