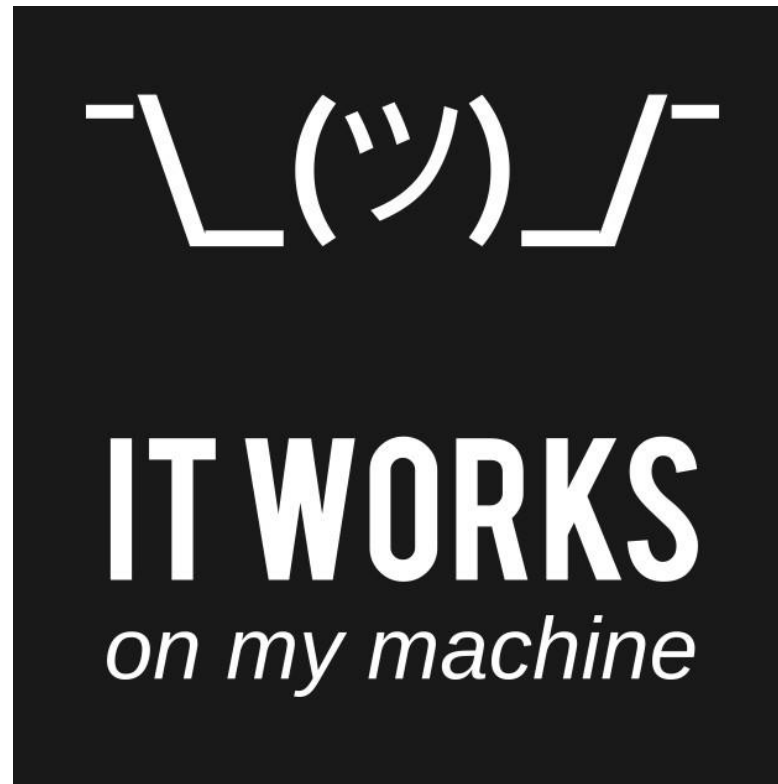


# Docker for Reproducibility



Kristin Chen

# Agenda

- Why do we need containers?
- How can containers solve those problems?
- What is Docker container?
  - How does Docker containize?
  - What is Docker architecture?
- Demo: running r analysis in container
- Resources and usage at Westat
- Reference

# Why do we need containers?

- Scenario 1: rerun *my\_analysis\_12042018\_KC.rmd* but get an error: *“Error in xx: could not find function”xx*
  - You then figure out the function *xx* is deprecated
- Scenario 2:
  - You send *analysis\_12042019\_KC.rmd* to your teammate, along with the *private library* stored in the project directory created by *packrat* to manage package dependencies.
  - A few minutes later, your teammate pins you on Skype and says he encountered an error: *“package ‘X’ is not available (for R version 3.3.0)”*.
  - You two then figure out it’s because the R version in your PC is 3.6.0, which is compatible for package X; however, your teammate has R version 3.3.0 installed in his PC.

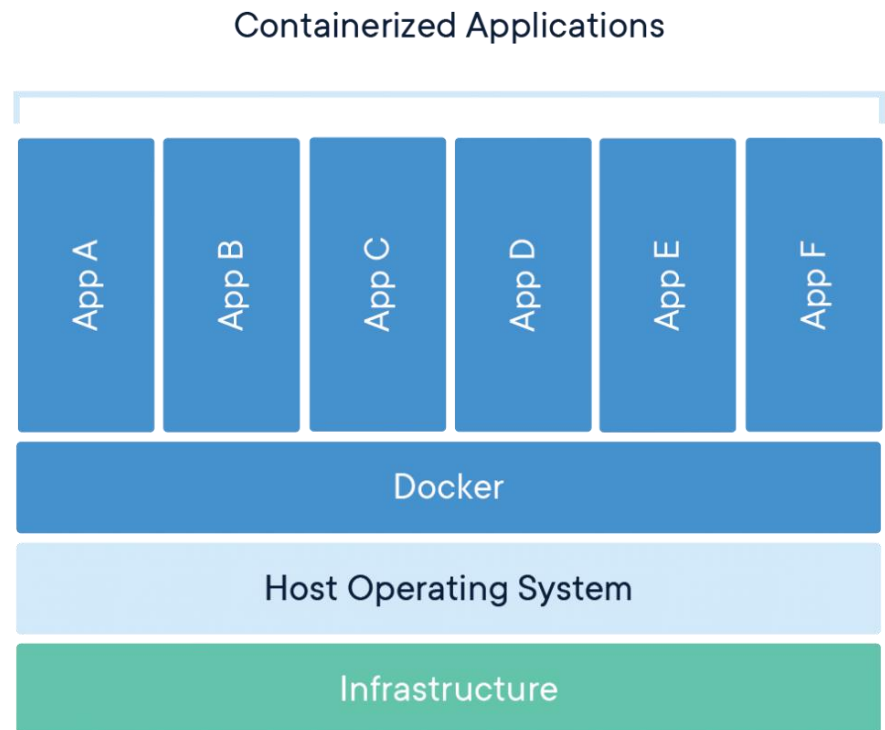
# Why do we need containers? (cont.)

- challenge:
  - the computation environment where the analyses build is **different** from where the analyses execute
  - replicating prior analysis or allowing other users to reproduce your analysis in their machines become troublesome.
    - computation environment, in the context of analysis in R programming, is included functions in the packages, packages, and R version, etc.



# How can containers solve those problems?

- *containerize* the computation environment, or the **infrastructure** of your analyses as code
- run each analysis/service with its own dependencies in separate containers, ensuring same results replicate on **any** computer environment
- isolate from the main computation environment in that PC



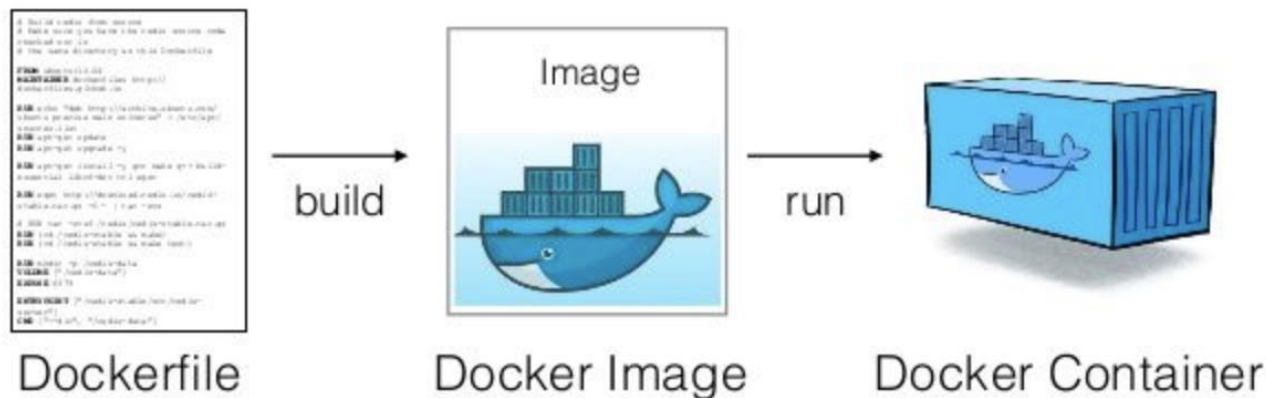
# What is Docker container?

- Docker: shipping companies, which carry containers from one part of the world to another
  - no containers:
    - everything needs to be transported are individually loaded into the ship
    - loading and unloading of goods difficult
  - with containers:
    - everything can be loaded or unloaded quickly using cranes
- **analysis** is *good* in the story above
  - analysis: a collection of computation pieces included packages, system dependencies, codes, reports
- Docker creates a “**containized**” version of the analysis
  - everything needed to run the analysis is included
  - allow reproducing the analysis whenever on Linux, Mac or Windows.



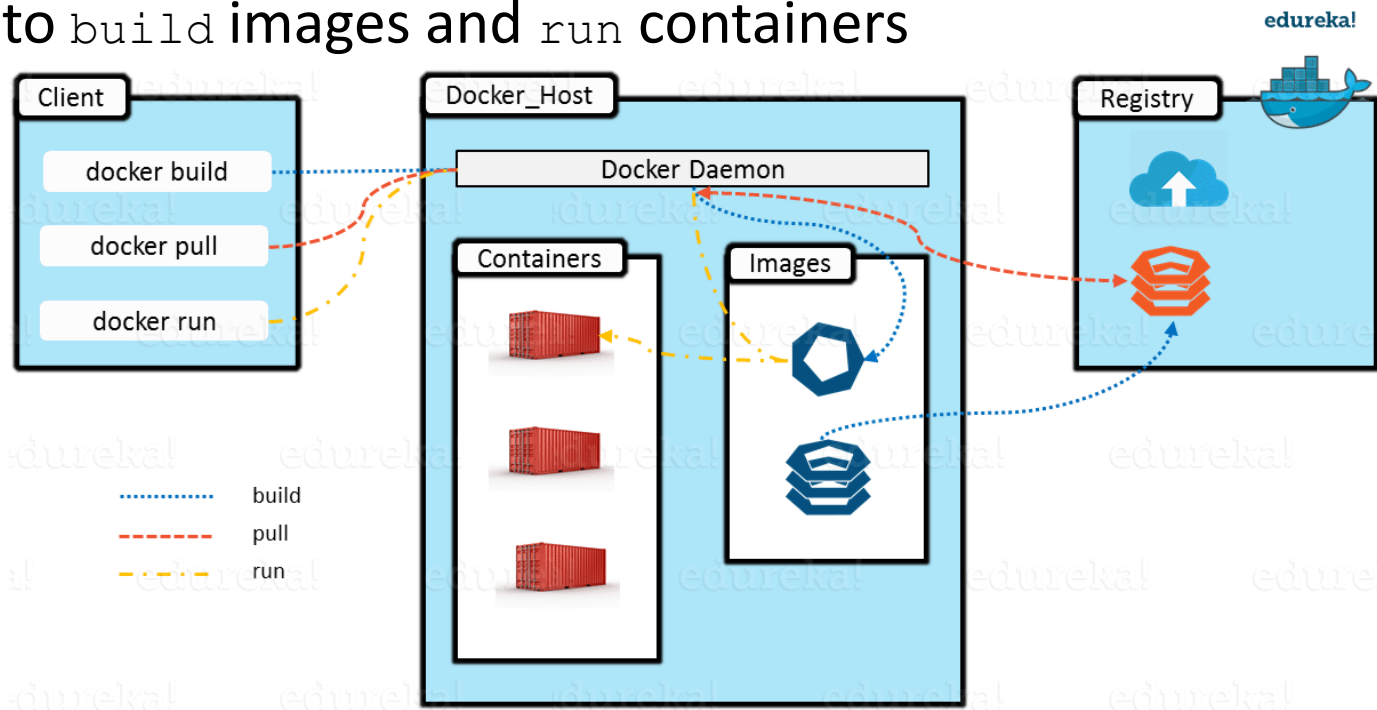
# What is Docker container - how does Docker containerize?

- *build* an image from Dockerfile, and *spin up* a running container from that image
  - Docker image: a template of your analyses/applications computation environment
  - Docker container: a running instance of Docker image
  - Dockerfile: a list of commands in a special text file to create an image



# What is Docker container - what is Docker architecture?

- Docker registry: a central repository for versioning images.
  - DockerHub is a well-known public registry where you could push or pull Docker images
- Docker CLI: where interact with Docker host using command lines to `build` images and `run` containers





# Demo: running r analysis in container

- repo: <https://github.com/jiatingchen/docker-for-reproducibility>
- step:
  - `docker build Dockerfile`
  - `docker run image-nme`
  - render *my\_analysis.rmd* to *my\_analysis.html* in container
  - copy *my\_analysis.html* to local machine
- walk through Dockerfile:
  - rocker: <https://www.rocker-project.org/>
  - FROM RUN COPY CMD
  - differences between computation env in container vs that in my PC

```
library(e1071)
library(caret)
sessionInfo()
## R version 3.6.0 (2019-04-26)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## other attached packages:
## [1] caret_6.0-81      ggplot2_3.2.1     lattice_0.20-38  e1071_1.7-0.1
```
- build repo in binder

# Resources and usage at Westat

- gitlab
- private Docker registry
- DevOps

# Reference

<https://reproducible-analysis-workshop.readthedocs.io/en/latest/8.Intro-Docker.html>

<https://arxiv.org/pdf/1410.0846.pdf>

<https://hackernoon.com/5-free-online-courses-to-learn-docker-for-beginners-492cfc488ecb>

<https://cloudblogs.microsoft.com/opensource/2019/07/15/how-to-get-started-containers-docker-kubernetes/>

<https://nickjanetakis.com/blog/understanding-how-the-docker-daemon-and-docker-cli-work-together>

<https://ropenscilabs.github.io/r-docker-tutorial/>

<https://machinelearningmastery.com/machine-learning-in-r-step-by-step/>

<https://colinfay.me/docker-r-reproducibility/>

Thank you!

