2023 ARM/ASR Joint Meeting - Open Science for ARM and ASR

Reproducibility in Data Analysis with Docker

Insights from the Sage Project

Bhupendra A. Raut, Sean Shahkarami, Yongho Kim, Bobby Jackson,
Nicola Ferrier, Pete Beckman and Scott Collis

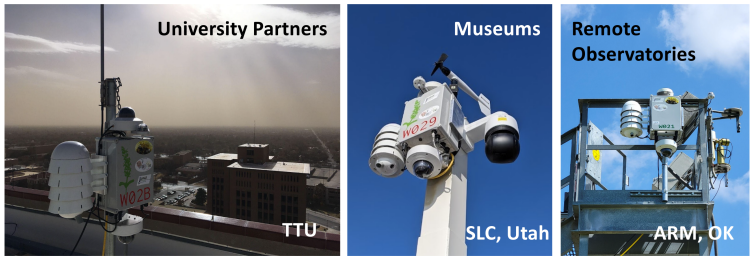
Northwestern-Argonne Institute of Science and Engineering, Northwestern University, Evanston, IL
Argonne National Laboratory, Lemont, IL

09 August 2023

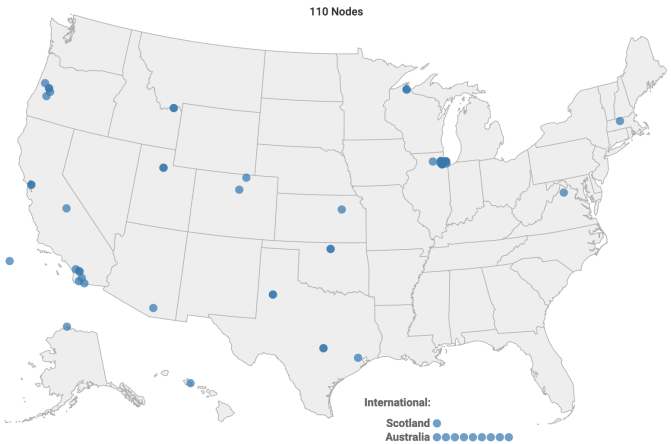
Northwestern University

Argonne NATIONAL LABORATORY

Running AI@Edge with Sage infrastructure



Blade System

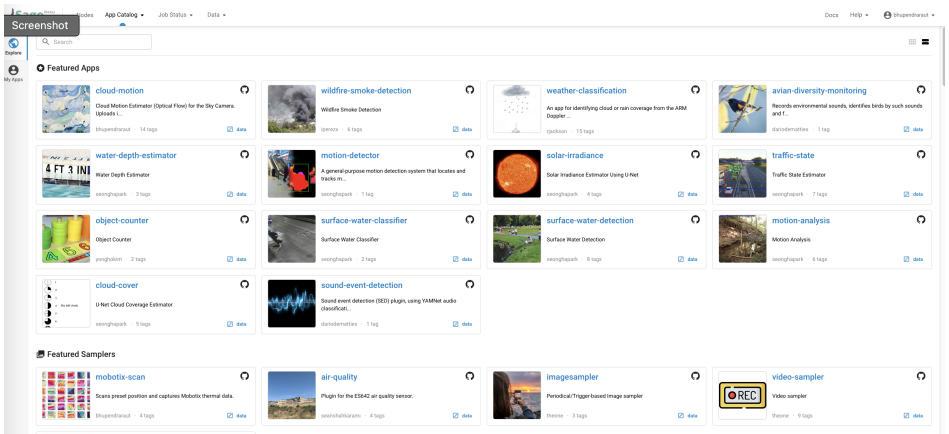


Nodes
110

Active Jobs
104

Recent Apps
25

App Data
2,029,895
Records in the last 24 hours*



sagecontinuum.org/

Reproducibility in Sage Application Deployment

When we develop an edge app (Beckman, et al., 2016) we need

1. to set up the libraries with different versions.
2. same setup across different systems.
3. same setup across different projects or developers.
4. track changes in the setup with time.

Question

How do we reproduce exactly the required environment for each app in every node?

Docker

Docker is an open-source and user-friendly tool created for easy software deployment by process isolation and environment-independent computing without the overhead of a full-fledged virtual machine.

Figure 1.3. A basic computer stack running two programs that were started from the command lin

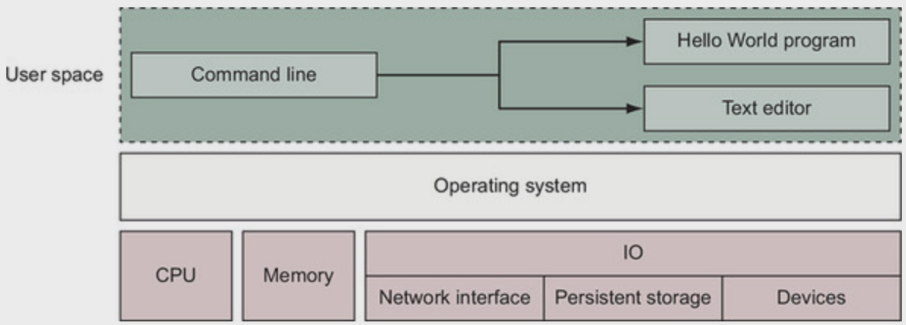
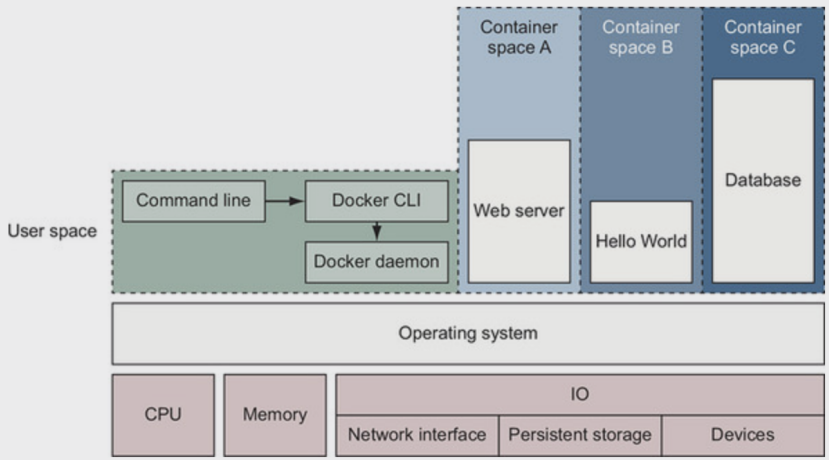


Figure 1.4. Docker running three containers on a basic Linux computer system

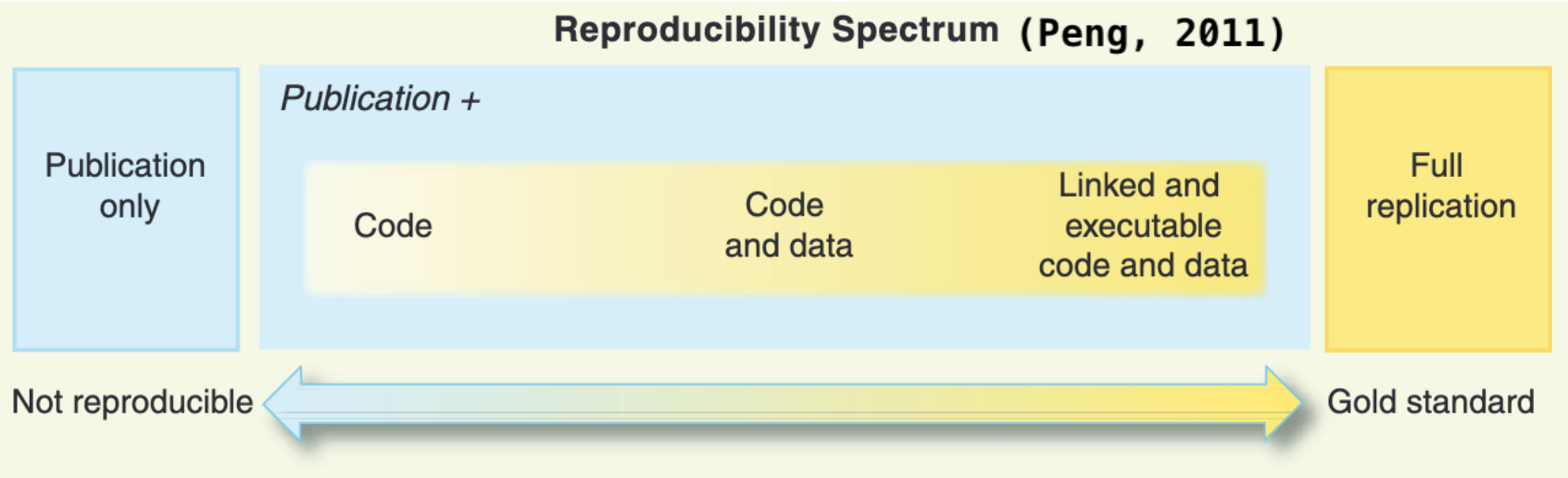


Docker in action (Nickoloff and Kuenzli, 2019)

In terms of scope and isolation:

Virtual Machines > Docker > Dependency Managers

Reproducibility in Research



Reproducible Analysis $\approx$ Data + Code + Environment

- Open data are essential for open and reproducible research.
- Git and github are widely accepted tools for open code.
- But the code is alive only in its compatible environment.

You want to share your work with the world and with your future self.

Same Question

How would you ensure that your code environment is reproducible?

Application with Docker

```
#ARM-ASR 2023

# Print "Hello, world!"
message = "Hello, world!"
print(message)

# Save the message to a text file
with open("/data/message.txt", "w") as file:
    file.write(message)
```

We will containerize this application using Docker.

Basics of Docker: 1. Images and Containers

Docker Image

A Docker image is an immutable, layered, and versioned snapshot that encapsulates software, dependencies, and configurations. It is a standalone executable software package.

Build images with `docker build -t rbhupi/arm-asr2023 .`

If you have docker installed you can get this image with

`docker pull rbhupi/arm-asr2023`

Dockerhub link for the image

Docker Container

A Docker container is a runnable instance of a Docker image.

Run like an executable with `docker run rbhupi/arm-asr2023`

Basics of Docker: 2. Dockerfile and ENTRYPOINT

Dockerfile

A Dockerfile is a text file containing instructions for building a Docker image.

vim Dockerfile

```
FROM python:3.8.2-alpine

RUN pip3 install opencv-python

RUN mkdir /data/
COPY ./* /app/


WORKDIR /app
ENTRYPOINT ["python3", "/app/app.py"]
```

While running the container, you can override the ENTRYPOINT using

`docker run -it --entrypoint /bin/sh rbhupi/arm-asr2023`

```
(base) ➜  demo1 git:(main) ✗ docker run  rbhupi/arm-asr2023
hello world !
(base) ➜  demo1 git:(main) ✗ docker run -it --entrypoint /bin/sh  rbhupi/arm-asr
2023
/app # ls
Dockerfile  app.py
/app # ls /
app     data    etc     lib     mnt     proc    run     srv     tmp     var
bin     dev     home    media   opt     root    sbin    sys     usr
/app # exit
(base) ➜  demo1 git:(main) ✗
```

Basics of Docker: 3. Connecting to the Host Machine

Port Mapping

Mapping ports between the host system and containers to enable communication.

Use the '-p' flag with the docker run command

```
docker run -p host_port:container_port image_name
```

Volume Mounting

Sharing data between host and containers using volume mounts.

Set the '-v' flag

```
docker run -v /host/path:/container/path image_name
```

Basics of Docker: 3. Connecting to the Host Machine

```
docker run -v /host/path:/container/path image_name
```

```
(base) ➜  demo1 git:(main) ✗ ls /Users/bhupendra/temp
(base) ➜  demo1 git:(main) ✗ docker run  -v /Users/bhupendra/temp:/data/ rbhupi/
arm-asr2023
Hello, world!
(base) ➜  demo1 git:(main) ✗ ls /Users/bhupendra/temp
message.txt
(base) ➜  demo1 git:(main) ✗ more /Users/bhupendra/temp/message.txt
Hello, world!
(base) ➜  demo1 git:(main) ✗
```

Dockerhub link for the image

Sage

A Software-Defined Sensor Network

Cyberinfrastructure for Edge Computing

www.sagecontinuum.org

ARM JOINT MEETING 2023

ASR Atmospheric System Research

Northwestern University · The University of Chicago · Northern Illinois University · Argonne National Laboratory · Colorado State University · UC San Diego · George Mason University · Array of Things · ARM · NEON National Ecological Observatory Network · HPWREN · SDSC San Diego Supercomputer Center · Lincoln Park Zoo For Wildlife. For All.

Sage

References

- Beckman, P., Sankaran, R., Catlett, C., Ferrier, N., Jacob, R. and Papka, M., 2016. Waggle: An open sensor platform for edge computing. In IEEE SENSORS.
- Nickoloff, J. and Kuenzli, S., 2019. Docker in action. Simon and Schuster.
- Peng, R.D., 2011. Reproducible research in computational science. Science, 334(6060), pp.1226-1227.