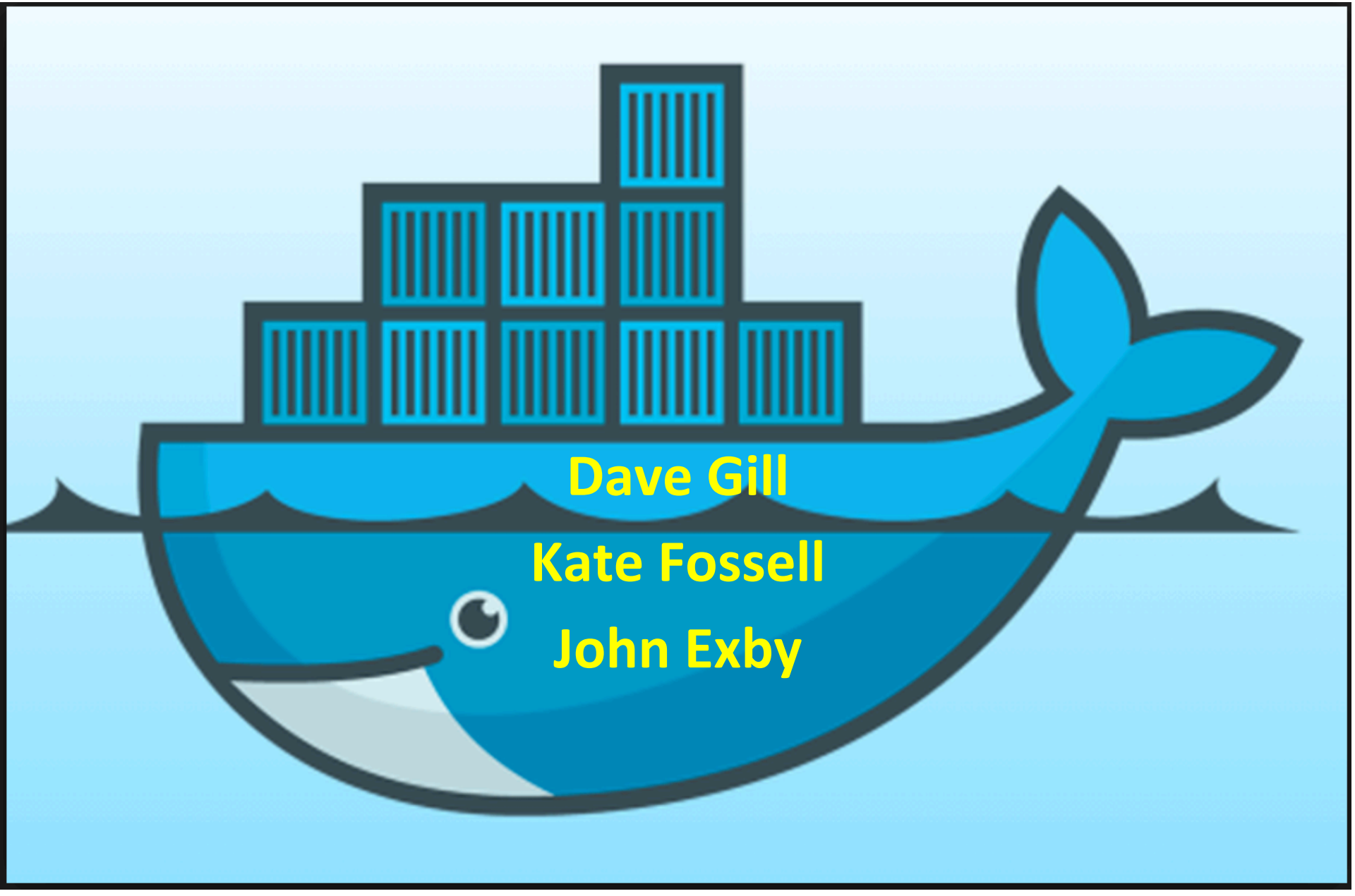
# WRF in a Docker Container



# Ye Olde Outline

- **What is a docker container**
- **Why do I want WRF inside a container**
- **How do I get this WRF docker thing**
- **How do I use this WRF docker thing**
- **More of the DEMONSTRATION**

# What is a docker container?

- A docker container
  - is open source
  - is an easy way to build a development environment
  - can hold applications “inside the container”
  - is portable across Linux, Mac, and Windows machines
  - is much smaller than a virtual machine
  - sets up a user-defined partition between the host machine and “container land”
  - allows “root” inside the container, but does not alter permissions on the host machine
  - works across a single node

# Why do I want WRF in a container?


Software containers can revolutionize research and education with numerical weather prediction models by easing use and guaranteeing reproducibility.

Hacker et al., 2017

http://journals.ametsoc.org/doi/pdf/10.1175/BAMS-D-15-00255.1

# Why do I want WRF in a container?

- For ***beginning WRF users***, the ability to use the complicated NWP model is hampered by the complexity of the state-of-the-art system
  - Libraries: netcdf, MPI, grib compression
  - Executables: netcdf, ncl, mpi, Centos pieces, ssh
  - Static data
  - Grib data
  - No place to conduct online WRF tutorials
  - No admin support on laptops
  - If the code base (data, directories) are destroyed, how is the system reconstructed

# How do I get the WRF docker container?

- Download the tar file from the WRF Tutorial page
- This has three files (really small, total < 20 kB)
  - Dockerfile
  - Configuration file with a really long name
  - README.txt
- README.txt has simple instructions to follow

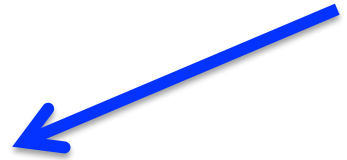
- **You need to have already SUCCESSFULLY installed docker on your target machine (your laptop)**

# How do I use the WRF docker container?

- Does the docker “hello, world” test work?

- Make a docker directory
- Put the Dockerfile and config file in that directory
- Build the WRF container (about 5 min)

```
>> docker build -t wrf_tutorial .
```



- Run the WRF container (about 30 seconds)

```
>> docker run -it --name teach_me \
wrf_tutorial /bin/tcsh
```

# How do I use the WRF docker container?

- What if I want to get data in or out of the container?

```
>> docker run -it --name teach_me \
-v some_localhost_dir:/wrf/wrfoutput \
wrf_tutorial /bin/tcsh
```

- The local directory (***some_localhost_dir***) and the container directory (**/wrf/wrfoutput**) are mapped to each other, can “see” each other

# How do I use the WRF docker container?

- How do I get back inside the same container if I “exit”ed out?

```
>> docker start -ai teach_me_ext
```

# How do I use the WRF docker container?

- Some useful commands
  - Container – an instance, from “run” command

```
>> docker ps -a
>> docker rm teach_me
```

- Some useful commands
  - Images – immutable, from “build” command

```
>> docker images -a
>> docker rmi wrf_tutorial
```

# How do I use the WRF docker container?

- There are a couple build idiosyncrasies
- WRF (configure.wrf)
  - After the configure step, add a flag to help openmpi

```
DM_CC                   =               mpicc -DMPI2_SUPPORT
```

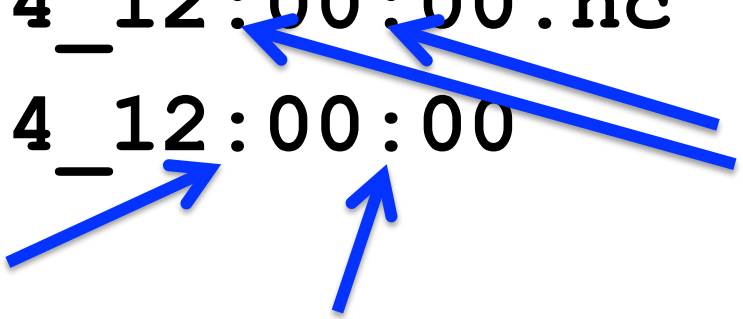
- WPS (configue.wps)
  - The netcdff library needs to be added to WRF_LIB

```
-L$(NETCDF)/lib CONFIGURE_NETCDFF_LIB \
-lnetcdf -lnetcdff
```

# How do I use the WRF docker container?

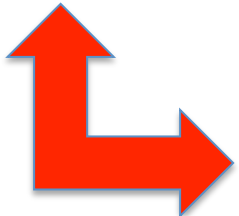
- If your **hosting OS** is windows – be careful with the **colons** that are in the WRF system file names

```
met_em.d01.2000-01-24_12:00:00.nc
wrfout_d01_2000-01-24_12:00:00
```



- These are problematic when placed in the visible volume that the local windows host can see

```
&time_control
 nocolons = F
/
```



Affects only model output

Other files must be manually modified