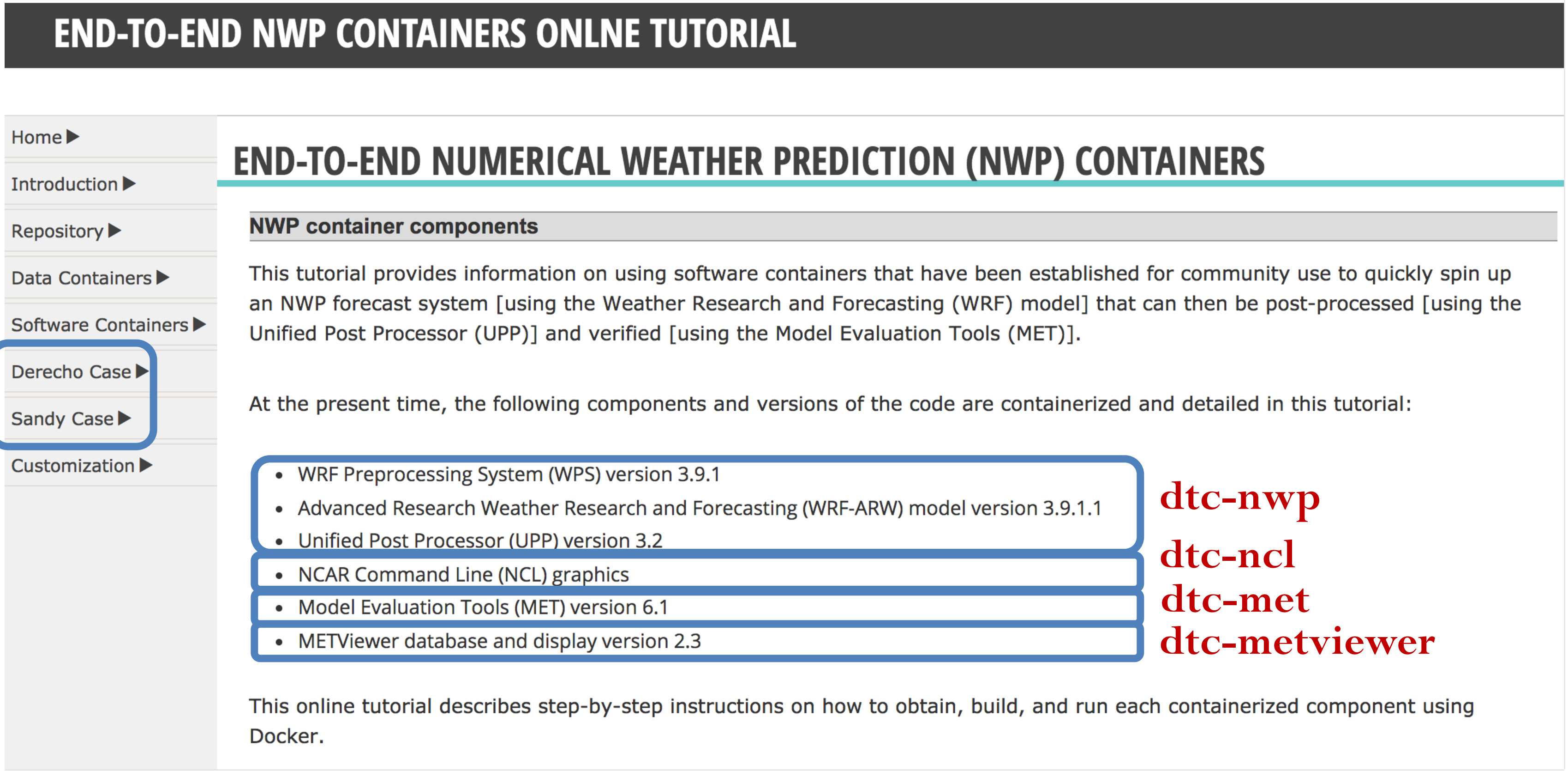
Docker Containers

Container NWP Tutorial

https://dtcenter.org/met/docker-nwp/tutorial/container_nwp_tutorial/index.php



END-TO-END NWP CONTAINERS ONLNE TUTORIAL

END-TO-END NUMERICAL WEATHER PREDICTION (NWP) CONTAINERS

NWP container components

This tutorial provides information on using software containers that have been established for community use to quickly spin up an NWP forecast system [using the Weather Research and Forecasting (WRF) model] that can then be post-processed [using the Unified Post Processor (UPP)] and verified [using the Model Evaluation Tools (MET)].

At the present time, the following components and versions of the code are containerized and detailed in this tutorial:

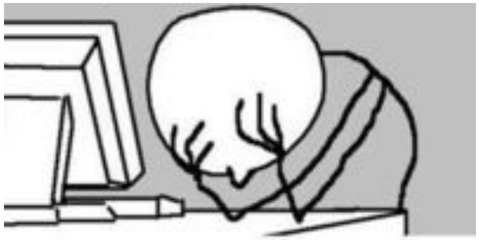
- WRF Preprocessing System (WPS) version 3.9.1
- Advanced Research Weather Research and Forecasting (WRF-ARW) model version 3.9.1.1
- Unified Post Processor (UPP) version 3.2
- NCAR Command Line (NCL) graphics
- Model Evaluation Tools (MET) version 6.1
- METViewer database and display version 2.3

dtc-nwp
dtc-ncl
dtc-met
dtc-metviewer

This online tutorial describes step-by-step instructions on how to obtain, build, and run each containerized component using Docker.

Why use containers?

- Software systems require substantial set-up to get all the necessary code, including external libraries, compiled on a specific platform → ***Containers help solve this problem!***
  - Efficient, lightweight, secure, and self-contained (including operating system, libraries, code, and executables) systems
  - Everything required is packaged into isolated components, ready for development, shipment, and deployment directly to users
  - Software should always run the same, regardless of where it is deployed
  - Eliminates possible frustrations with up-front system setup



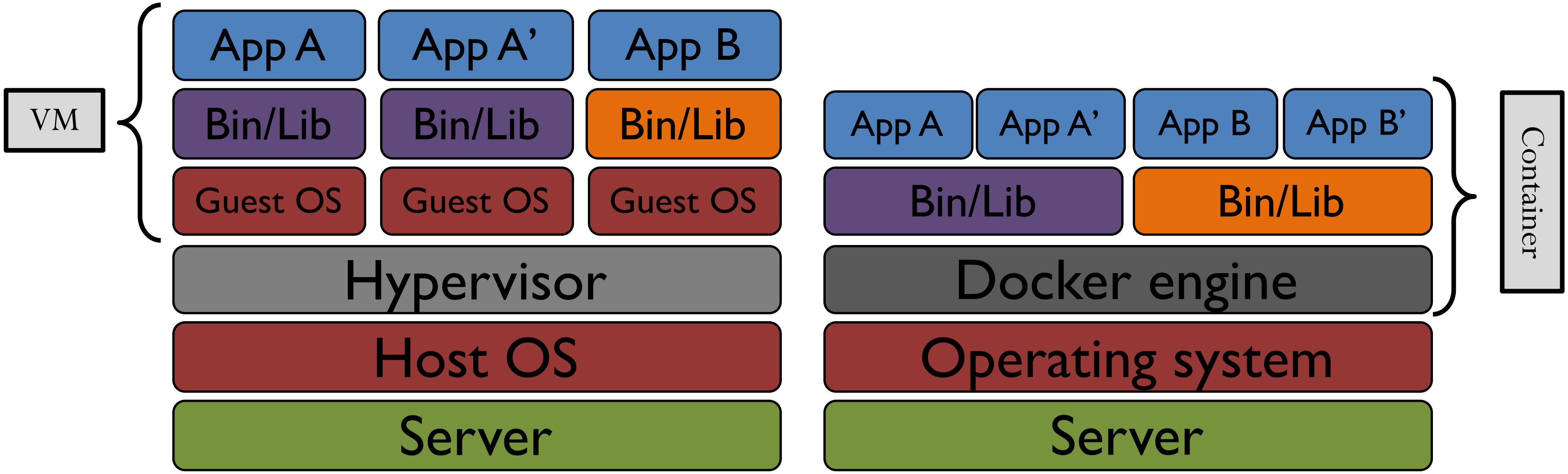
Someone trying to compile WRF

What is a Docker container?

- Docker is one of the leading software containerization platforms
  - Home page: **https://www.docker.com**
  - Documentation: **https://docs.docker.com**
- 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
  - ***requires root access to install Docker***
  - ***runs on a single node***

Virtual machine vs. containers

- Containers vs. virtual machines: VMs bundle a full operating system, whereas containers only contain necessary libraries and dependencies



Why use containers for NWP?

Advantages:

- ✓ Reduces spin-up time to build necessary code components
- ✓ Highly portable
- ✓ Use in cloud computing
- ✓ Easily sharable with other collaborators
- ✓ Easy to replicate procedures and results

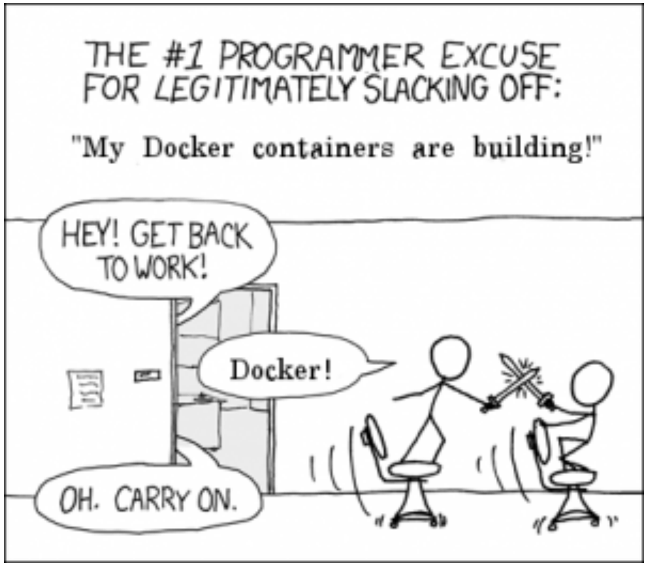
Who can benefit from containers?

- ✓ Graduate and undergrad students
- ✓ University faculty
- ✓ Researchers
- ✓ Tutorial participants

Ultimately, containers will substantially reduce the spin-up time with setting up software systems, which promotes greater efficiency in producing model and statistical output!

Images and Containers

- Images can be…
  - Built from scratch with a *Dockerfile (slower, but offers customization!)*
    - ***docker build –t dtc-met .***
  - Saved to a tar file, which can then be loaded for faster deployment
    - ***docker load –i dtc-met.tar.gz***

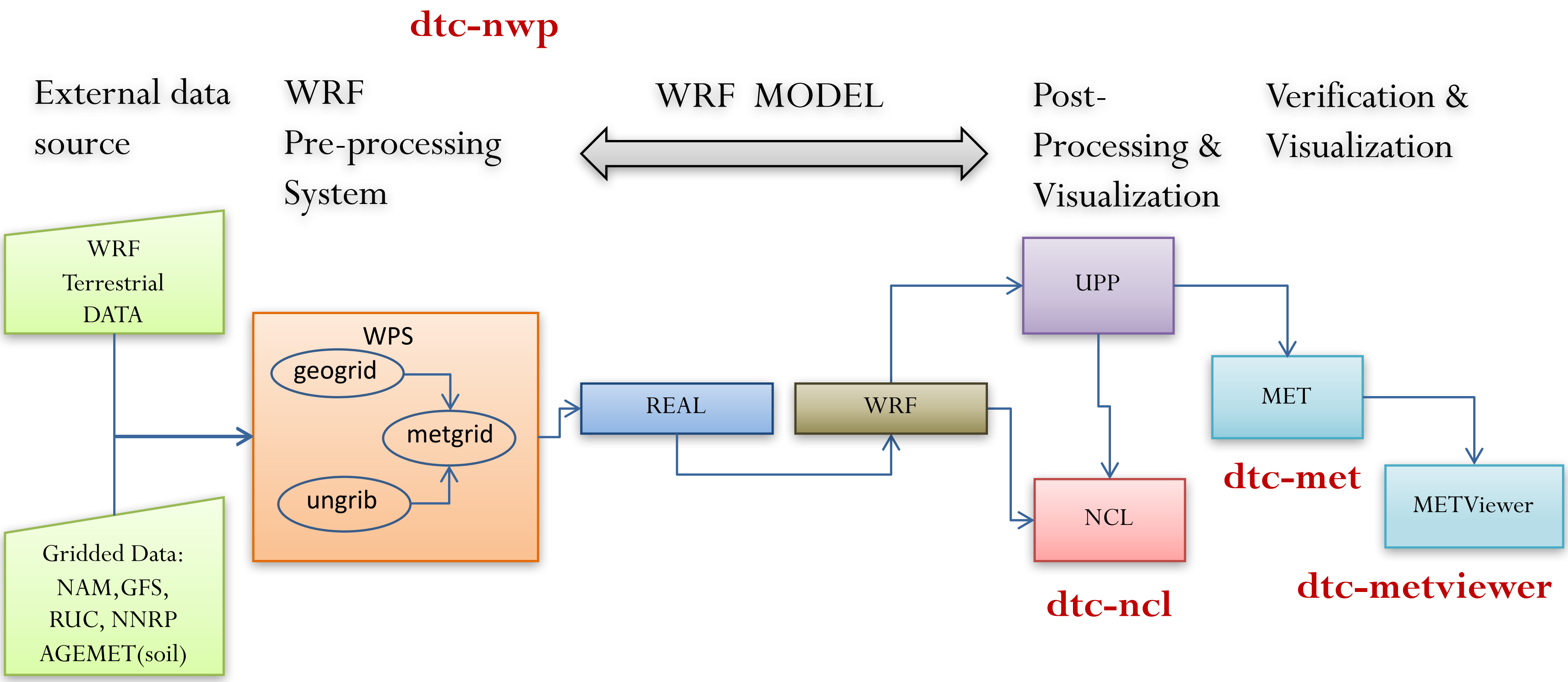

THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:
"My Docker containers are building!"
HEY! GET BACK TO WORK!
Docker!
OH. CARRY ON.

- Containers are…
  - An instance of an image
  - The result of docker run
    - ***docker run -it dtc-met /bin/bash***
  - The result of docker create
    - ***docker create --name wps_geog dtc-nwp-wps_geog***

Scripts and Configuration

- Shell scripts required to run each NWP component
- Necessary namelist and configuration files
  - Vtable.GFS
  - namelist.wps and namelist.input
  - MET configuration files
- Case-specific data
  - GFS files for ICs/LBCs
  - Observation data for gridded (Stage II) and point (NDAS prepbufr) verification
  - Sample METViewer plot xml

End-to-End NWP Workflow



dtc-nwp

External data source

WRF Pre-processing System

WRF MODEL

Post-Processing & Visualization

Verification & Visualization

WRF Terrestrial DATA

Gridded Data: NAM,GFS, RUC, NNRP AGEMET(soil)

WPS

geogrid

metgrid

ungrib

REAL

WRF

UPP

NCL

dtc-ncl

MET

dtc-met

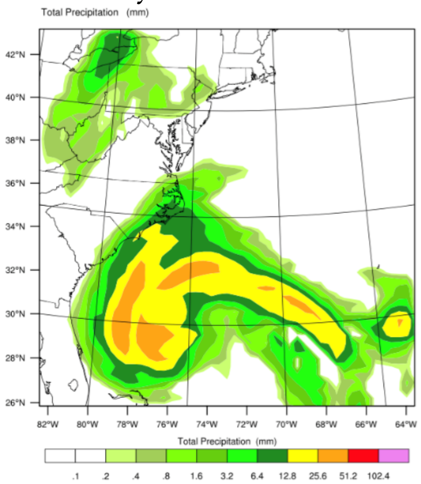
METViewer

dtc-metviewer

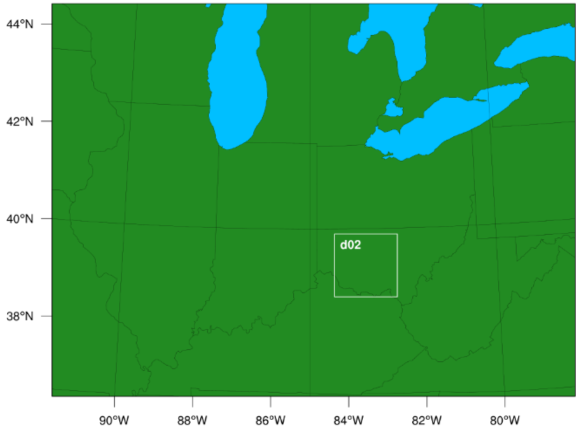
What is in the DTC containers? (1/2)

- DTC containers package everything that is needed to build and run the model and produce verification, including code and data
  - Uses gfortran; can be run serially or with distributed memory
- Two cases with full datasets are provided in current inventory
  - Hurricane Sandy (Initialized on 27 Oct. 2012)
    - 40-km domain centered over East Coast (6-h forecast)
  - Derecho event over the Eastern CONUS (Initialized on 29 June 2012)
    - 12-km parent domain with 3-km nest over southern Ohio (24-h forecast)

Total precip for Hurricane Sandy model forecast



Derecho case domain configuration

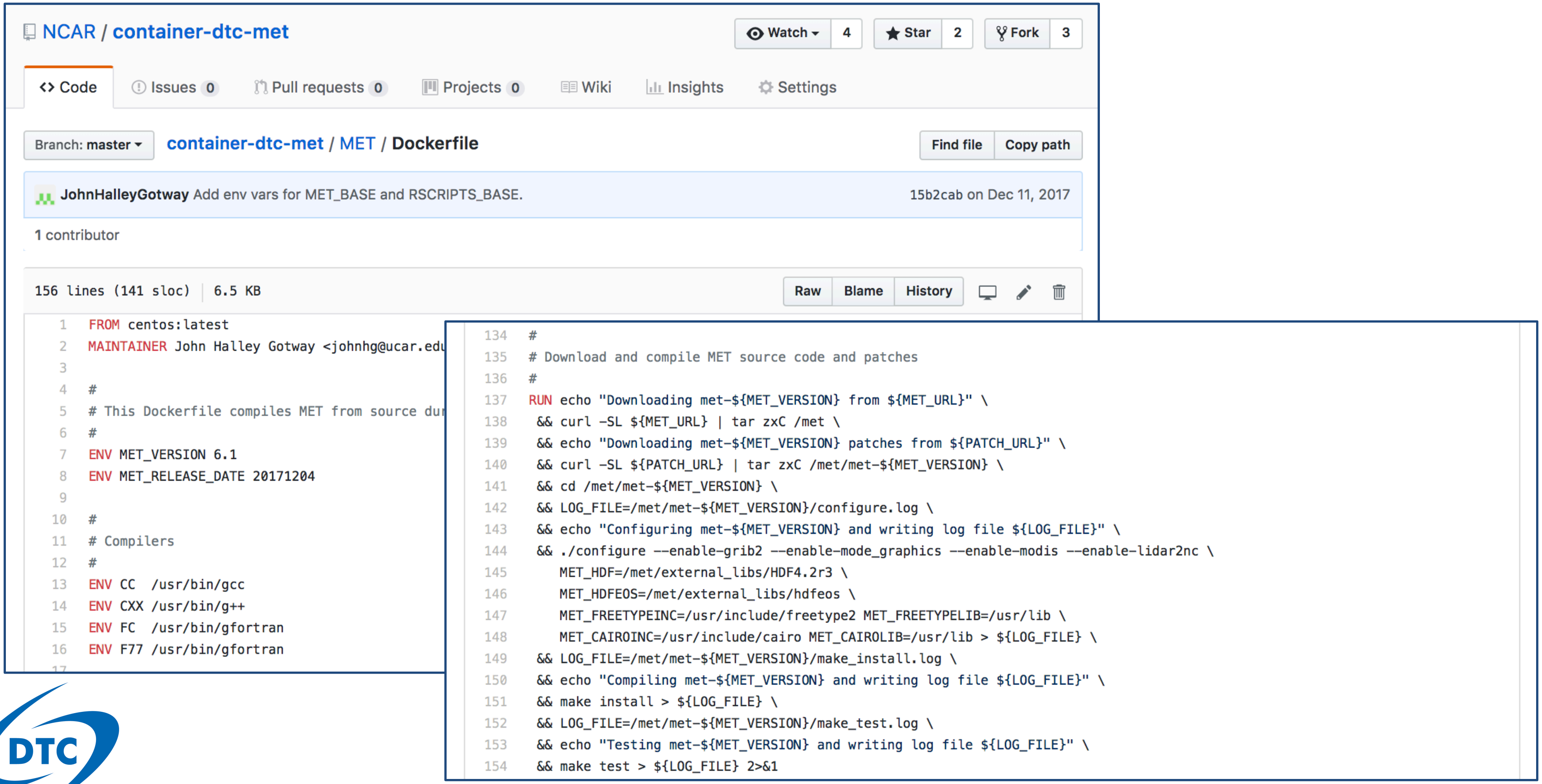


What is in the DTC containers? (2/2)

- README files with explicit instructions for building and running WPS, WRF, UPP, NCL, MET, and METViewer
- Necessary namelist and configuration files
  - Vtable.GFS
  - namelist.wps and namelist.input
  - MET configuration files
- Case-specific data
  - GFS files for ICs/LBCs
  - Observation data for gridded (Stage II) and point (NDAS prepbufr) verification
  - Sample METViewer XML plots

MET Container

- Dockerfile defines all steps necessary to compile MET from OS base image ~ ***docker build***



JohnHalleyGotway Add env vars for MET_BASE and RSCRIPTS_BASE. 15b2cab on Dec 11, 2017

1 contributor

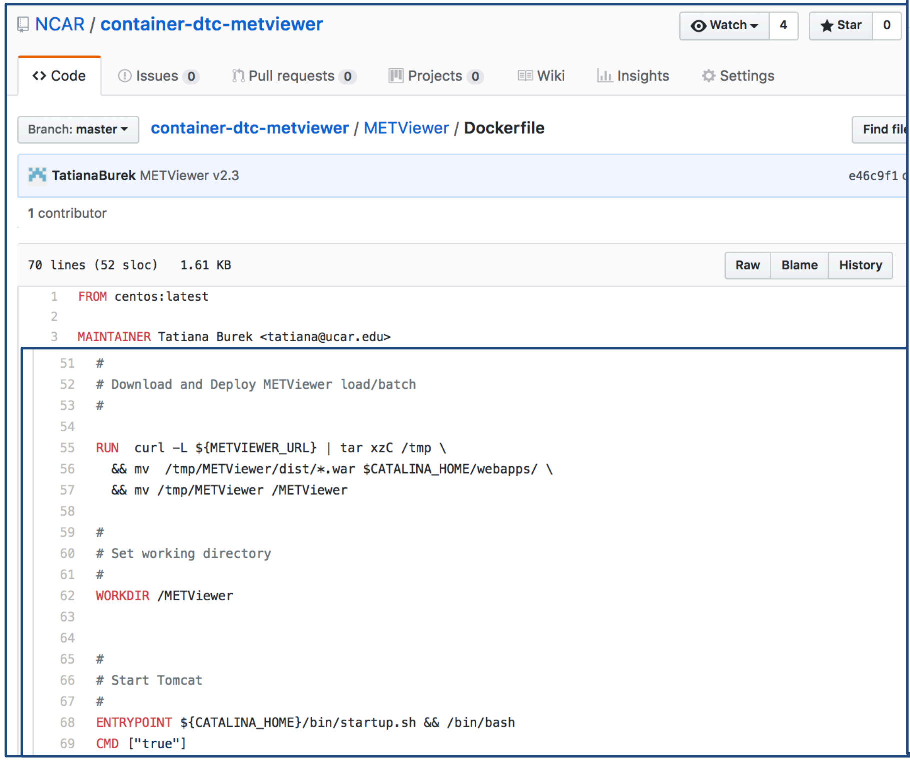
156 lines (141 sloc) | 6.5 KB

```
FROM centos:latest
MAINTAINER John Halley Gotway <johnhg@ucar.edu

#
# This Dockerfile compiles MET from source du
#
ENV MET_VERSION 6.1
ENV MET_RELEASE_DATE 20171204

#
# Compilers
#
ENV CC  /usr/bin/gcc
ENV CXX /usr/bin/g++
ENV FC  /usr/bin/gfortran
ENV F77 /usr/bin/gfortran
```

```
#
# Download and compile MET source code and patches
#
RUN echo "Downloading met-${MET_VERSION} from ${MET_URL}" \
 && curl -SL ${MET_URL} | tar zxC /met \
 && echo "Downloading met-${MET_VERSION} patches from ${PATCH_URL}" \
 && curl -SL ${PATCH_URL} | tar zxC /met/met-${MET_VERSION} \
 && cd /met/met-${MET_VERSION} \
 && LOG_FILE=/met/met-${MET_VERSION}/configure.log \
 && echo "Configuring met-${MET_VERSION} and writing log file ${LOG_FILE}" \
 && ./configure --enable-grib2 --enable-mode_graphics --enable-modis --enable-lidar2nc \
    MET_HDF=/met/external_libs/HDF4.2r3 \
    MET_HDFEOS=/met/external_libs/hdfeos \
    MET_FREETYPEINC=/usr/include/freetype2 MET_FREETYPELIB=/usr/lib \
    MET_CAIROINC=/usr/include/cairo MET_CAIROLIB=/usr/lib > ${LOG_FILE} \
 && LOG_FILE=/met/met-${MET_VERSION}/make_install.log \
 && echo "Compiling met-${MET_VERSION} and writing log file ${LOG_FILE}" \
 && make install > ${LOG_FILE} \
 && LOG_FILE=/met/met-${MET_VERSION}/make_test.log \
 && echo "Testing met-${MET_VERSION} and writing log file ${LOG_FILE}" \
 && make test > ${LOG_FILE} 2>&1
```

METViewer Container

- METViewer only available publicly as a container.
- Dockerfile defines all steps necessary to compile METViewer from OS base image ~ ***docker build***



NCAR / container-dtc-metviewer

container-dtc-metviewer / METViewer / Dockerfile

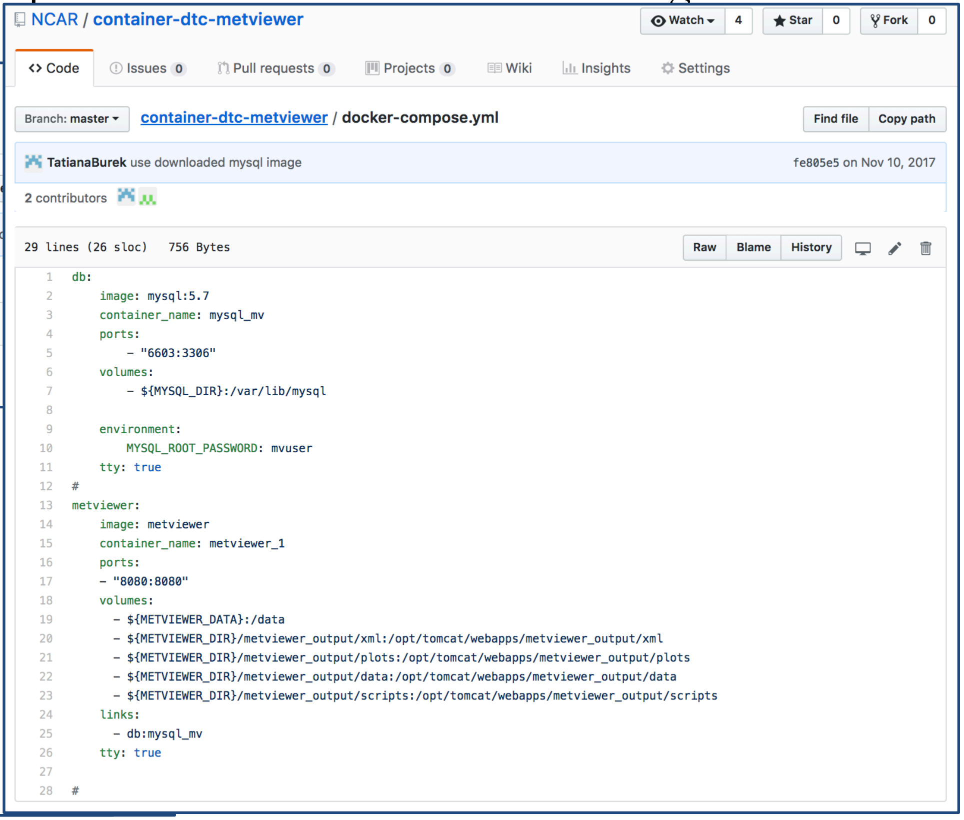
TatianaBurek METViewer v2.3 e46c9f1 c

1 contributor

70 lines (52 sloc) 1.61 KB

```
FROM centos:latest

MAINTAINER Tatiana Burek <tatiana@ucar.edu>
```

```
#
# Download and Deploy METViewer load/batch
#

RUN  curl -L ${METVIEWER_URL} | tar xzC /tmp \
  && mv  /tmp/METViewer/dist/*.war $CATALINA_HOME/webapps/ \
  && mv /tmp/METViewer /METViewer

#
# Set working directory
#
WORKDIR /METViewer


#
# Start Tomcat
#
ENTRYPOINT ${CATALINA_HOME}/bin/startup.sh && /bin/bash
CMD ["true"]
```



NCAR / container-dtc-metviewer

container-dtc-metviewer / docker-compose.yml

TatianaBurek use downloaded mysql image fe805e5 on Nov 10, 2017

2 contributors

29 lines (26 sloc) 756 Bytes

```
db:
    image: mysql:5.7
    container_name: mysql_mv
    ports:
        - "6603:3306"
    volumes:
        - ${MYSQL_DIR}:/var/lib/mysql

    environment:
        MYSQL_ROOT_PASSWORD: mvuser
    tty: true
#
metviewer:
    image: metviewer
    container_name: metviewer_1
    ports:
    - "8080:8080"
    volumes:
      - ${METVIEWER_DATA}:/data
      - ${METVIEWER_DIR}/metviewer_output/xml:/opt/tomcat/webapps/metviewer_output/xml
      - ${METVIEWER_DIR}/metviewer_output/plots:/opt/tomcat/webapps/metviewer_output/plots
      - ${METVIEWER_DIR}/metviewer_output/data:/opt/tomcat/webapps/metviewer_output/data
      - ${METVIEWER_DIR}/metviewer_output/scripts:/opt/tomcat/webapps/metviewer_output/scripts
    links:
      - db:mysql_mv
    tty: true

#
```

- Launch METViewer using METViewer and MySQL images ~ ***docker-compose***
- http://localhost:8080/metviewer/metviewer1.jsp

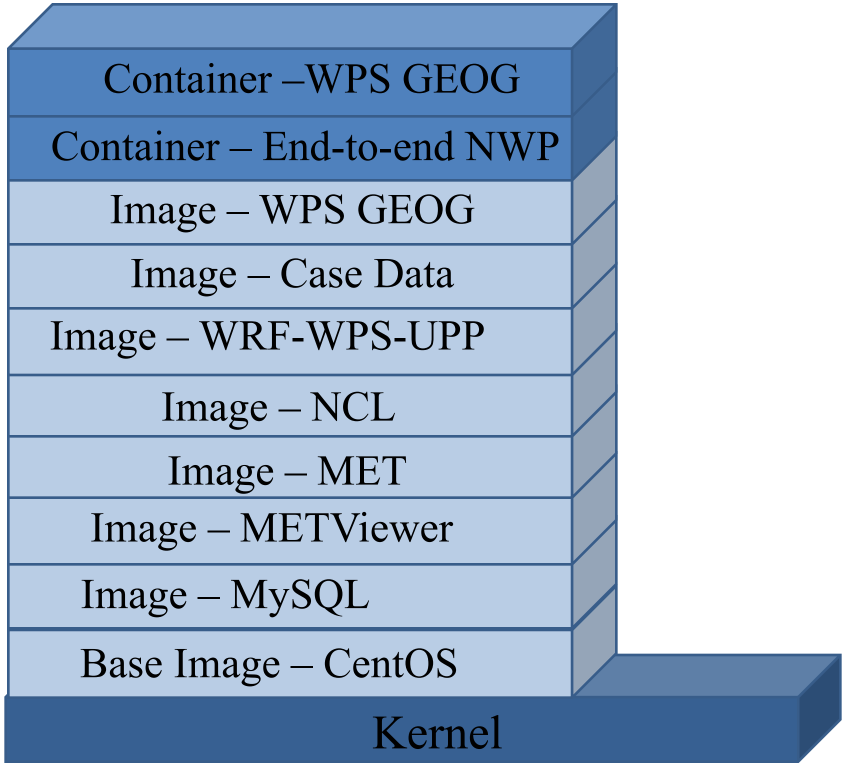
Helpful Docker commands

- docker build -t my-name . : builds image
- docker images : see what images are built
- docker rmi : remove image
- docker save my-name > my-name.tar.gz : save an image
- docker load < my-name.tar.gz : load a saved image

- docker run --rm -it --volumes-from -v --name : run a command in a new container
  - --rm: Automatically remove the container when it exits
  - -it: create an interactive bash shell in the container
  - --volumes-from: Mount volumes from the specified container(s)
  - -v: Bind mount a volume
  - --name: Assign a name to the container
- docker ps -a : see what containers are running; obtain container ID
- docker rm : remove container using ID

What does this look like for the end-to-end DTC container?

Contains necessary build and run commands to run end-to-end NWP workflow



Container –WPS GEOG

Container – End-to-end NWP

Image – WPS GEOG

Image – Case Data

Image – WRF-WPS-UPP

Image – NCL

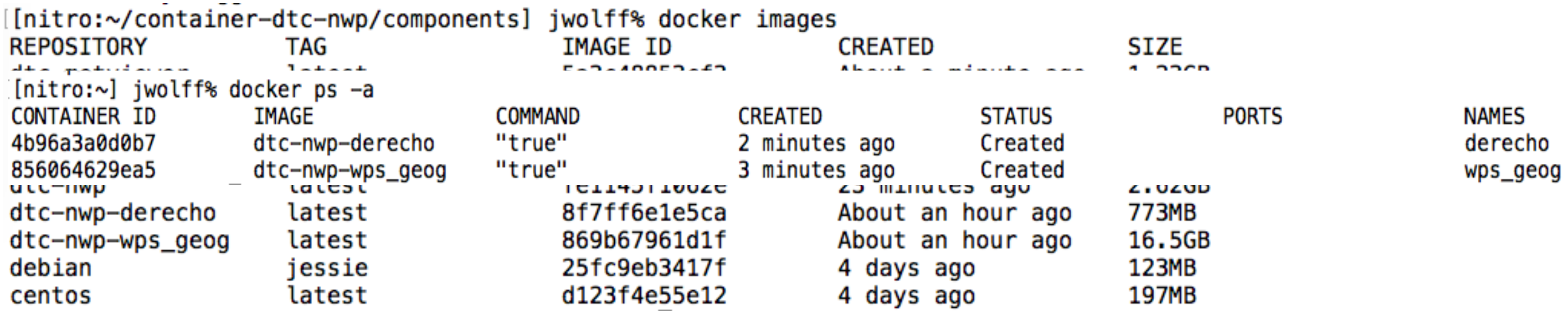
Image – MET

Image – METViewer

Image – MySQL

Base Image – CentOS

Kernel



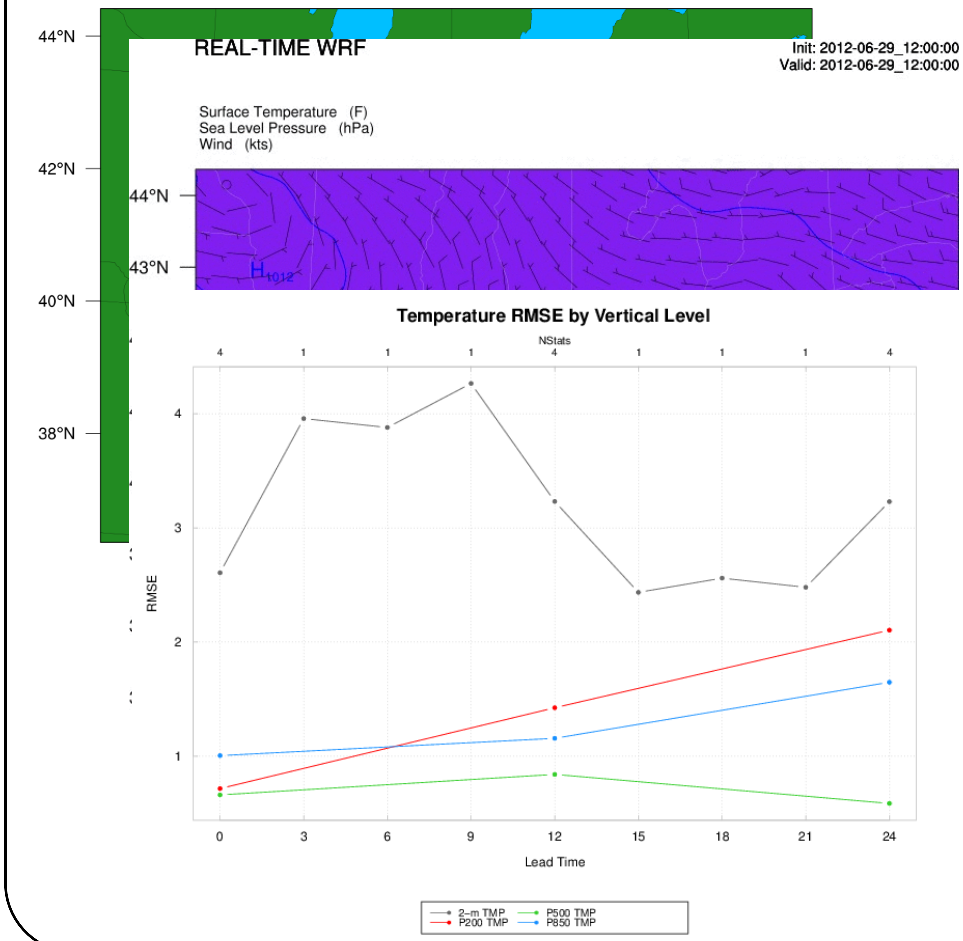
```
[nitro:~/container-dtc-nwp/components] jwolff% docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
```

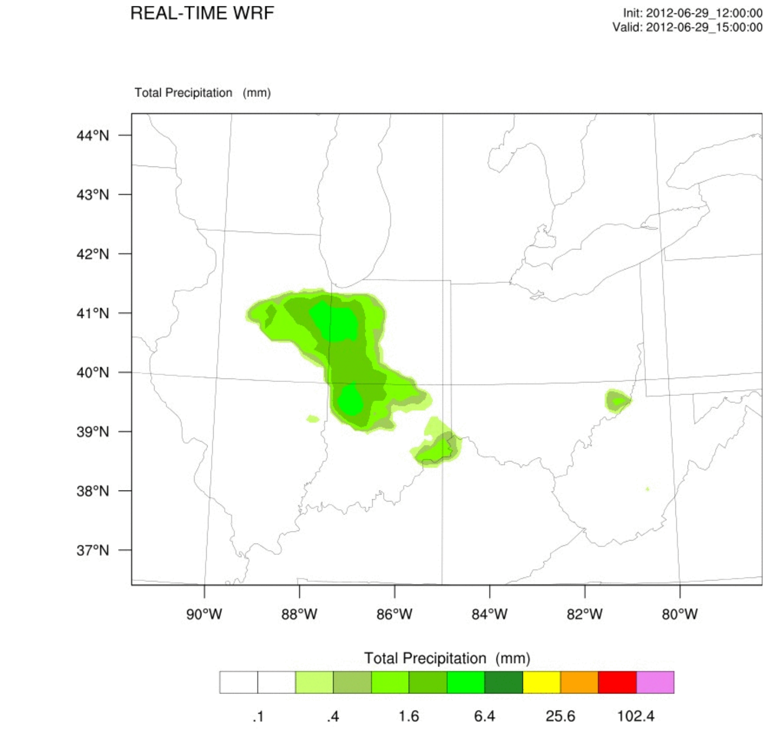
```
[nitro:~] jwolff% docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
4b96a3a0d0b7        dtc-nwp-derecho     "true"              2 minutes ago       Created                                 derecho
856064629ea5        dtc-nwp-wps_geog    "true"              3 minutes ago       Created                                 wps_geog
```

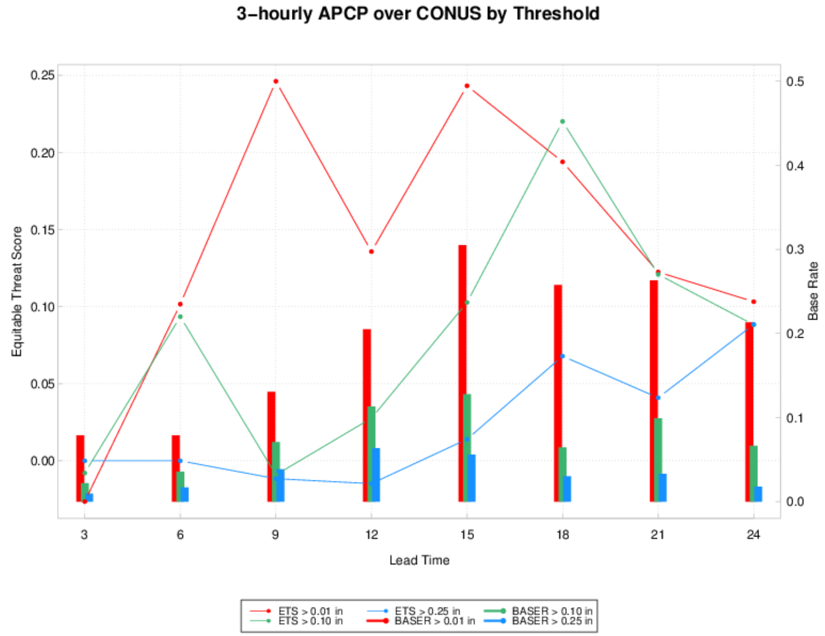
```
dtc-nwp-derecho     latest              8f7ff6e1e5ca        About an hour ago   773MB
dtc-nwp-wps_geog    latest              869b67961d1f        About an hour ago   16.5GB
debian              jessie              25fc9eb3417f        4 days ago          123MB
centos              latest              d123f4e55e12        4 days ago          197MB
```

Derecho Case

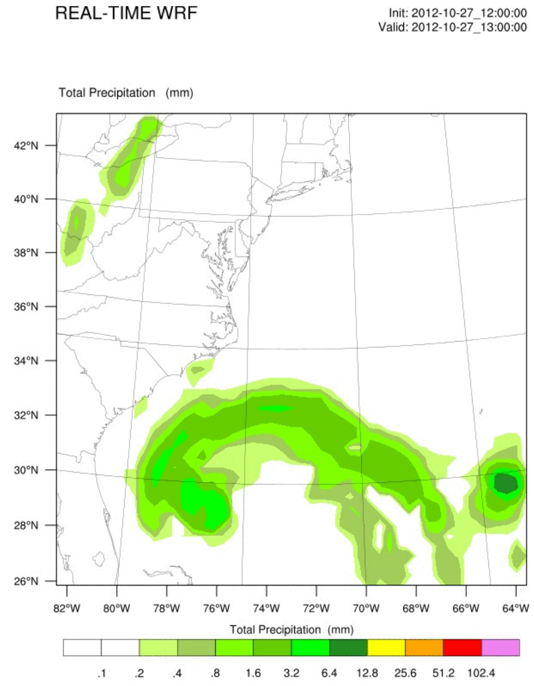
WPS Domain Configuration



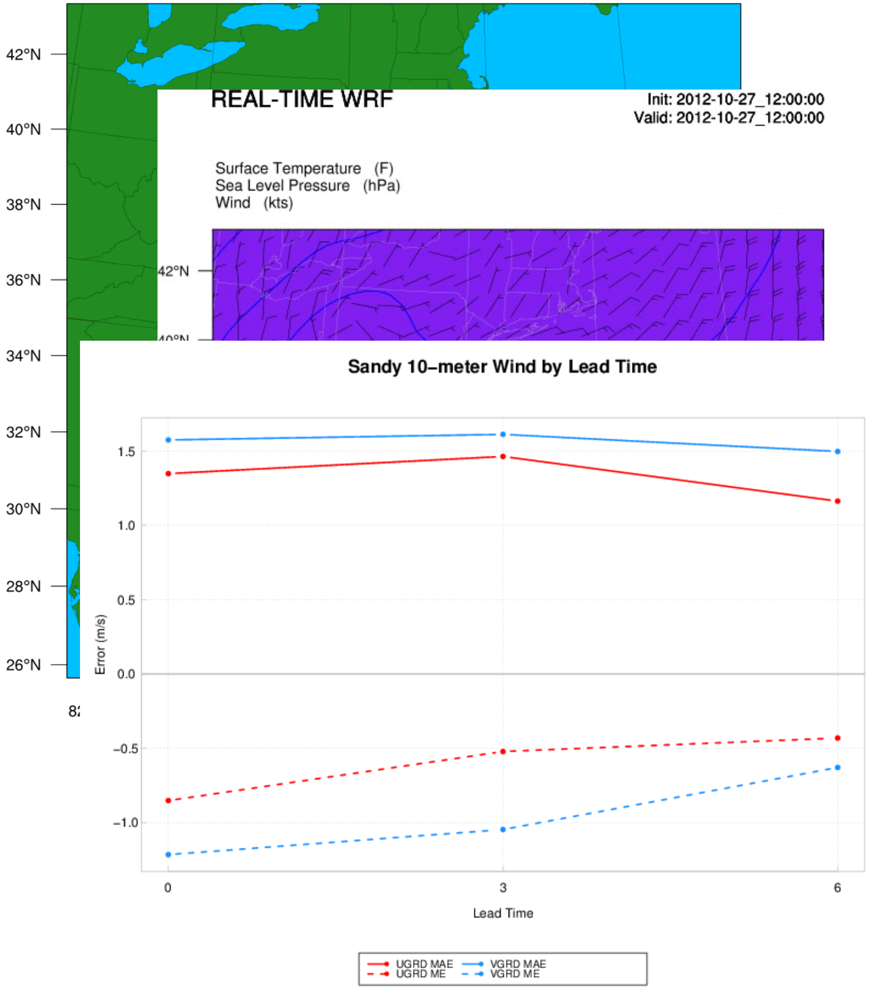


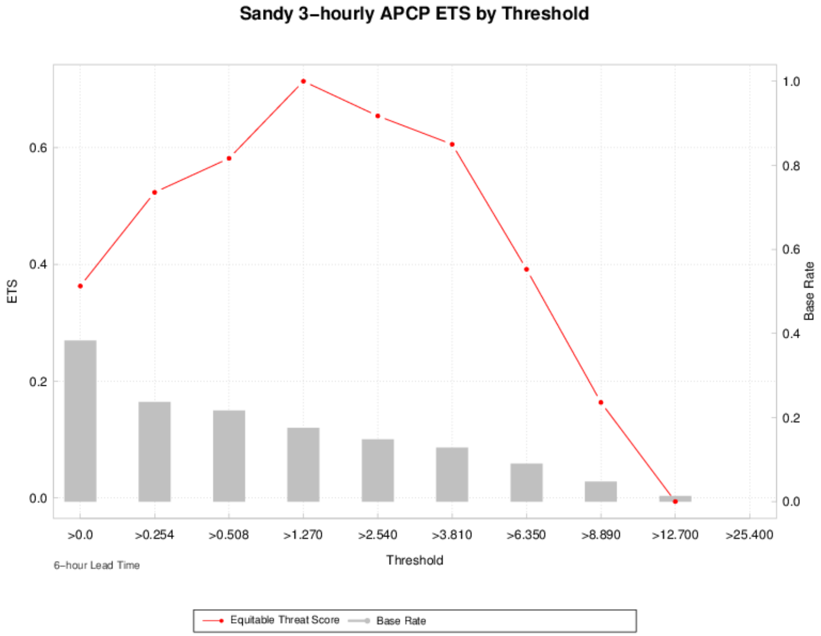


Sandy Case



WPS Domain Configuration





DTC Container Links

- **WRF, WPS, NCL, and UPP**
  - **https://github.com/NCAR/container-dtc-nwp/**
- **MET**
  - **https://github.com/NCAR/container-dtc-met/**
  - **https://dtcenter.org/met/users/downloads/docker_container.php** (Instructions for installing and running pre-built container)
- **METViewer (Containers for MySQL and METViewer)**
  - **https://github.com/NCAR/container-dtc-metviewer/**
- **End-to-end NWP container online tutorial**
  - **https://dtcenter.org/met/docker-nwp/tutorial/container_nwp_tutorial/index.php**

Future Work

- Expanded use for in-person tutorials
- Containerize Gridpoint Statistical Interpolation (GSI) data assimilation code
- Seek projects for using containers in cloud computing
- Deploy MET and METViewer containers to DockerHub
- Explore alternatives to Docker
  - Root access requirement is limiting
  - Issues mapping directories in Windows
  - Need to run with multiple nodes on HPC