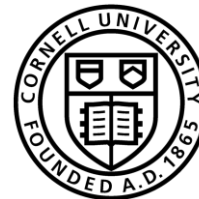




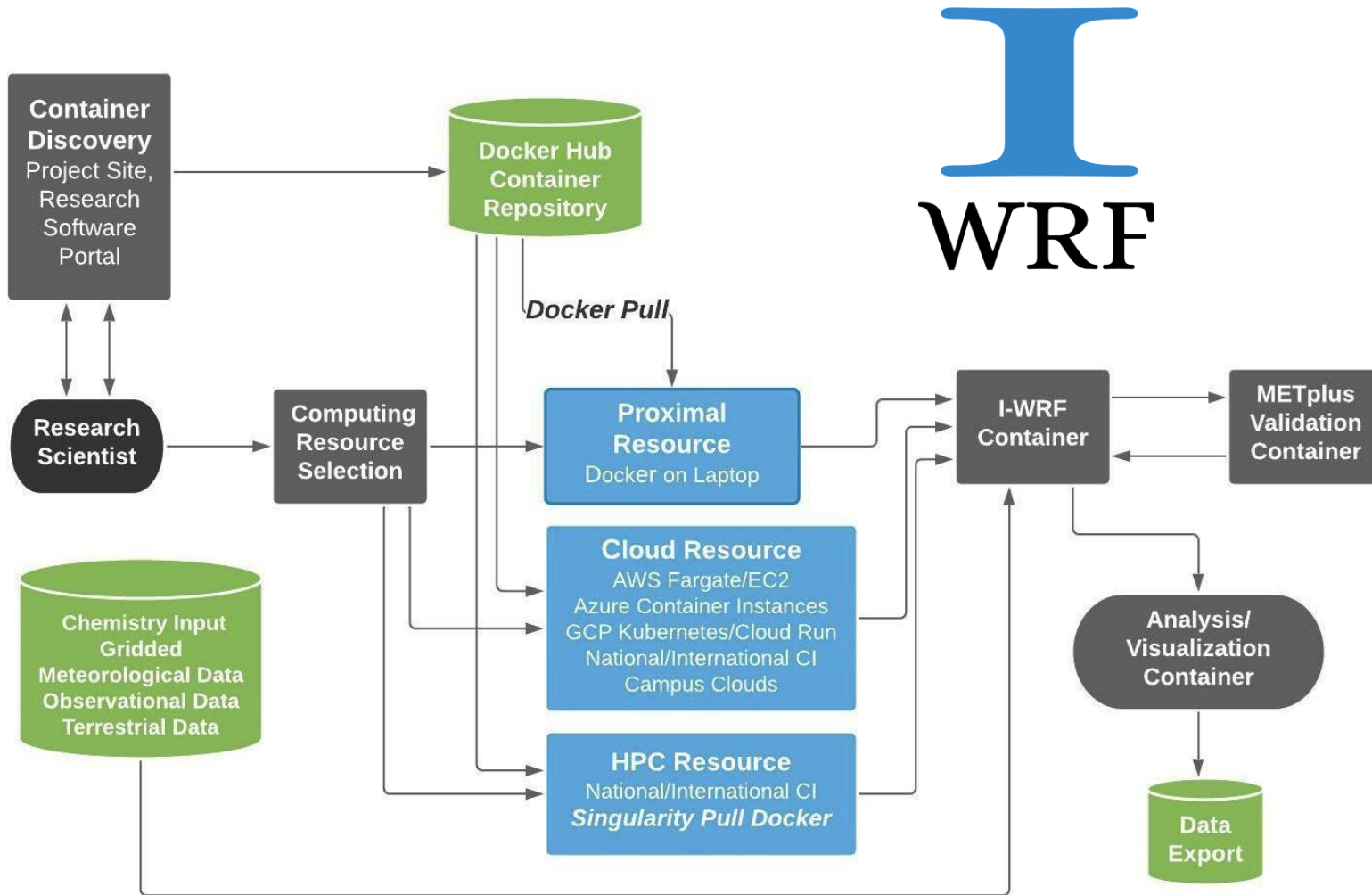
# Overview of I-WRF Container Architecture

Jared A. Lee & George McCabe (Presenters)  
NSF National Center for Atmospheric Research

(On behalf of the entire project team from NSF NCAR & Cornell)



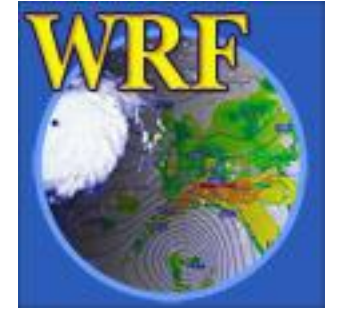
# I-WRF Conceptual Design



- The main change to this original concept diagram in our current implementation of I-WRF is that the visualization pieces are being done within the METplus validation container
- This way we only need to load in a Python environment in one container, not two

# WRF Challenges

- Despite a large user base and deployment on many HPC systems, around *50% of users* attending tutorials at NSF NCAR report difficulty configuring the software for use on whichever computing platform they're using
- Compiling WRF software requires understanding multiple compiler frameworks, a set of required libraries to be built with the same compiler you select for WRF, and a wide range of WRF configuration options
- Need to know where to obtain data for initial conditions & lateral boundary conditions (ICs/LBCs), and observations for verification
- It usually requires some work to get verification and visualization tools configured to ingest WRF output
- These technical barriers mean that potential researchers and scholars run into hurdles before they can even get to the weather and climate stuff



<https://www.istockphoto.com/signature/photo/thats-it-im-done-gm936117884-256071691>

# Verifying Model Output with METplus

- Model Evaluation Tools (METplus) verification system
  - Community validation toolkit supported by NSF NCAR and largely developed by NSF NCAR
  - Developed through funding from the 557<sup>th</sup> Weather Wing of the U.S. Air Force, National Oceanic and Atmospheric Administration (NOAA), and NSF NCAR
  - Verification framework that spans a wide range of temporal (warn-on-forecast to climate) and spatial (storm to global) scales
  - Used operationally by NOAA, UK Met Office, Australian Bureau of Meteorology, and others
  - Large community of users & contributors
- METplus was already containerized
- I-WRF containerizes METplus configurations for doing some sample verification from the the various I-WRF use cases, and plots it

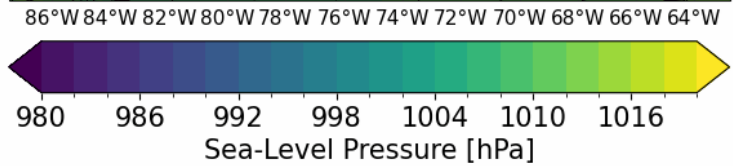
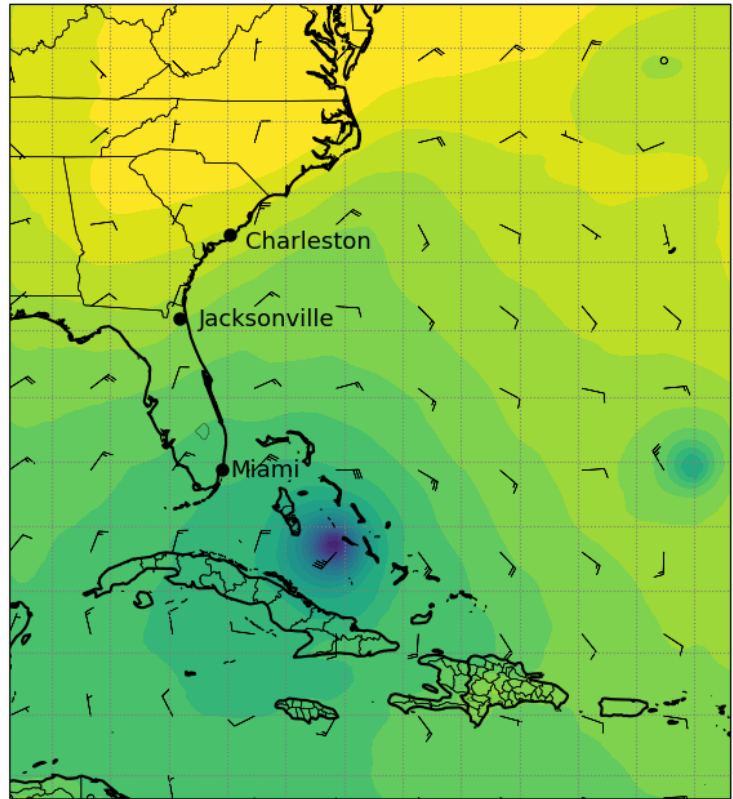
The logo for METplus, featuring the letters 'MET' in a large, bold, black sans-serif font, followed by 'plus' in a smaller, orange sans-serif font.

<https://dtcenter.org/community-code/metplus>

# I-WRF Hurricane Matthew Test Case Python Visualization

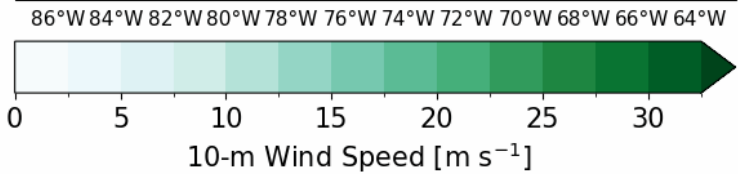
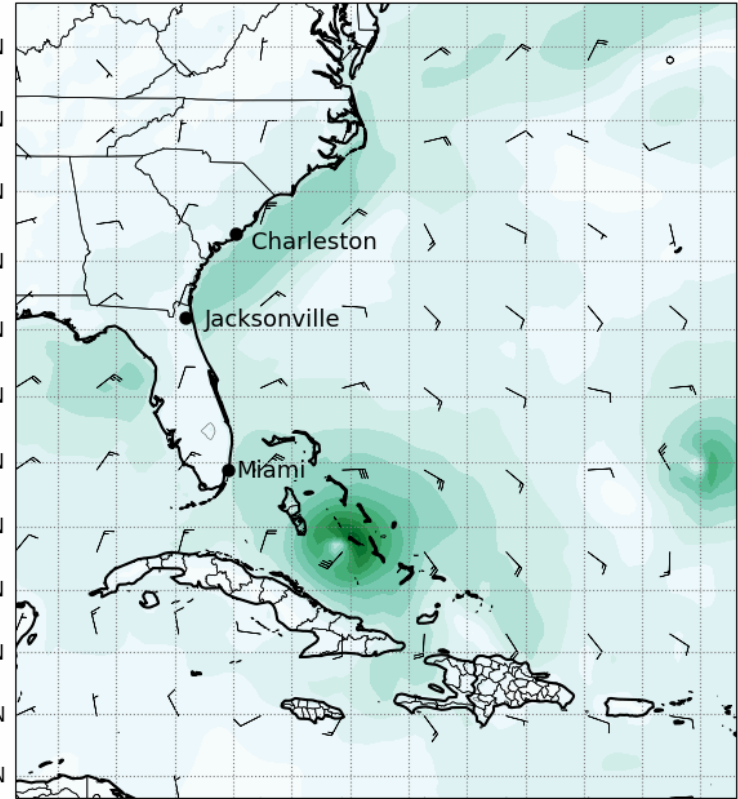
### Hurricane Matthew Test Case

Sea-Level Pressure      Start: 06 Oct 2016/0000 UTC  
Min: 982.0 hPa, Max: 1025.0 hPa Valid: 06 Oct 2016/0300 UTC  
86°W 84°W 82°W 80°W 78°W 76°W 74°W 72°W 70°W 68°W 66°W 64°W



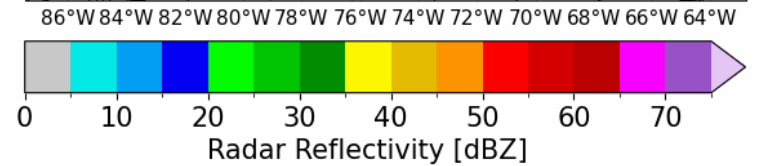
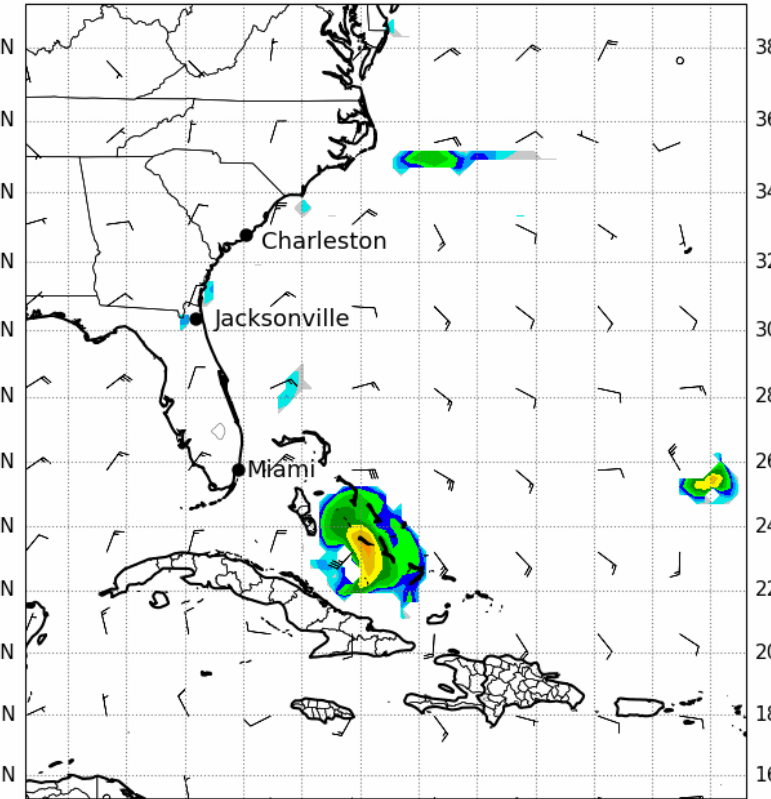
### Hurricane Matthew Test Case

10-m Wind Speed      Start: 06 Oct 2016/0000 UTC  
Min: 0.0 m s<sup>-1</sup>, Max: 31.1 m s<sup>-1</sup> Valid: 06 Oct 2016/0300 UTC  
86°W 84°W 82°W 80°W 78°W 76°W 74°W 72°W 70°W 68°W 66°W 64°W

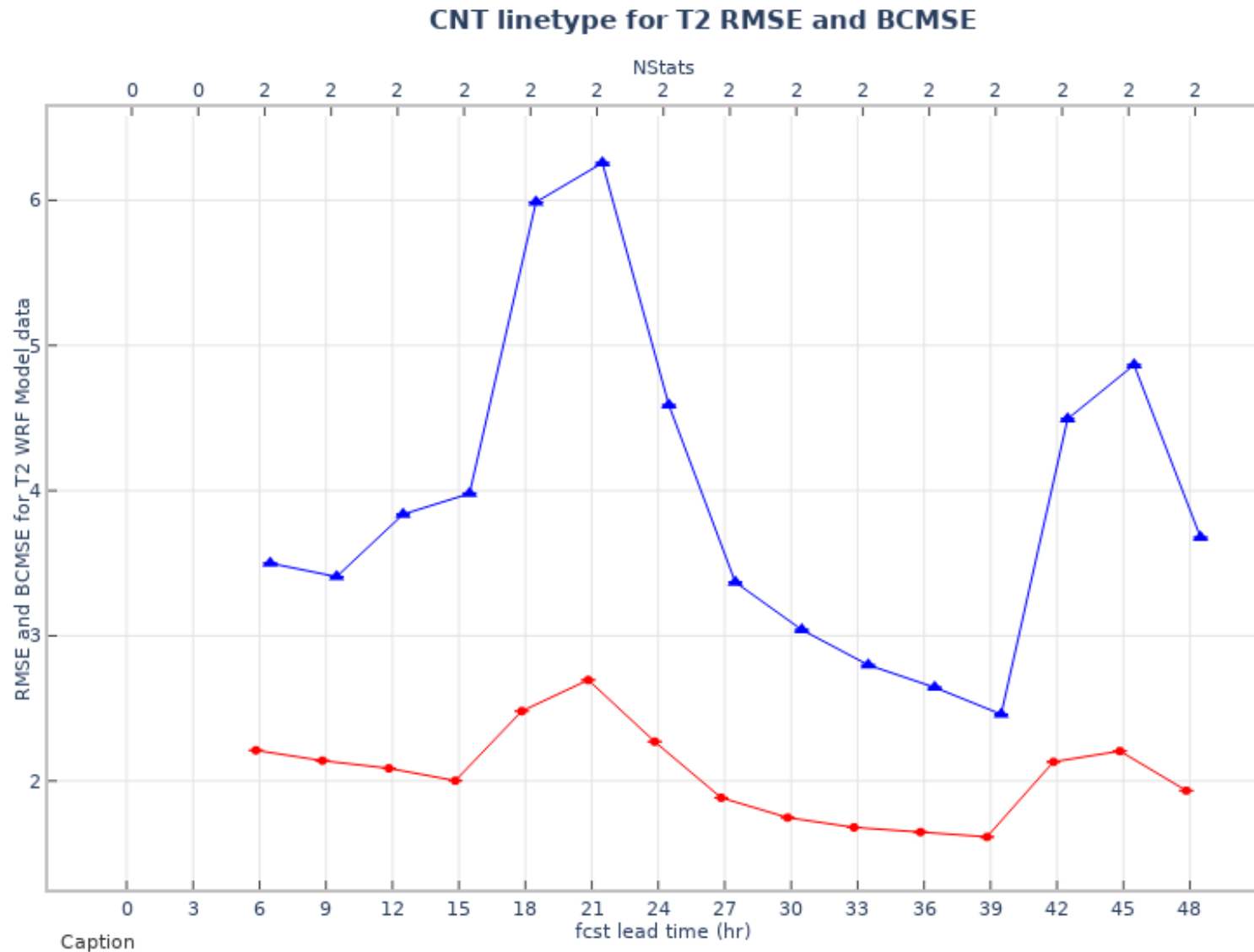


### Hurricane Matthew Test Case

Radar Reflectivity; 10-m Barbs      Start: 06 Oct 2016/0000 UTC  
Min: -30.0 dBZ, Max: 46.8 dBZ      Valid: 06 Oct 2016/0300 UTC  
86°W 84°W 82°W 80°W 78°W 76°W 74°W 72°W 70°W 68°W 66°W 64°W

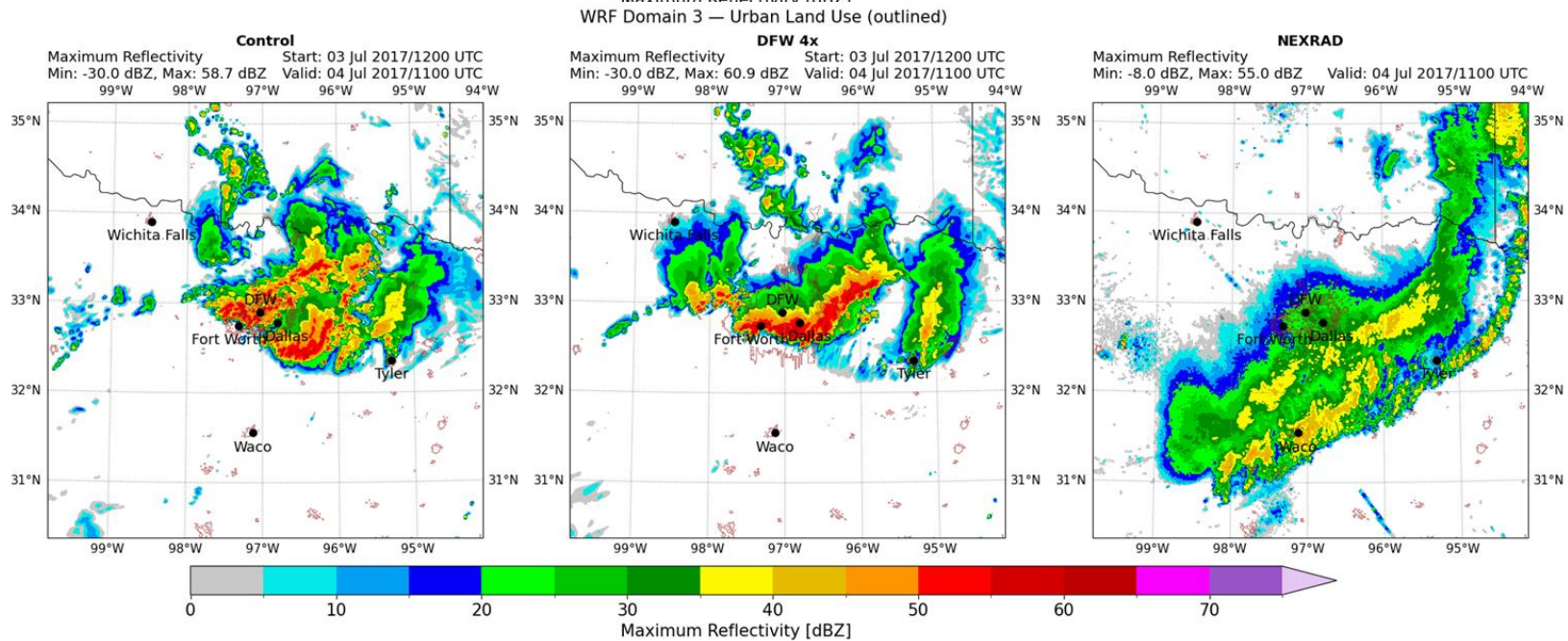
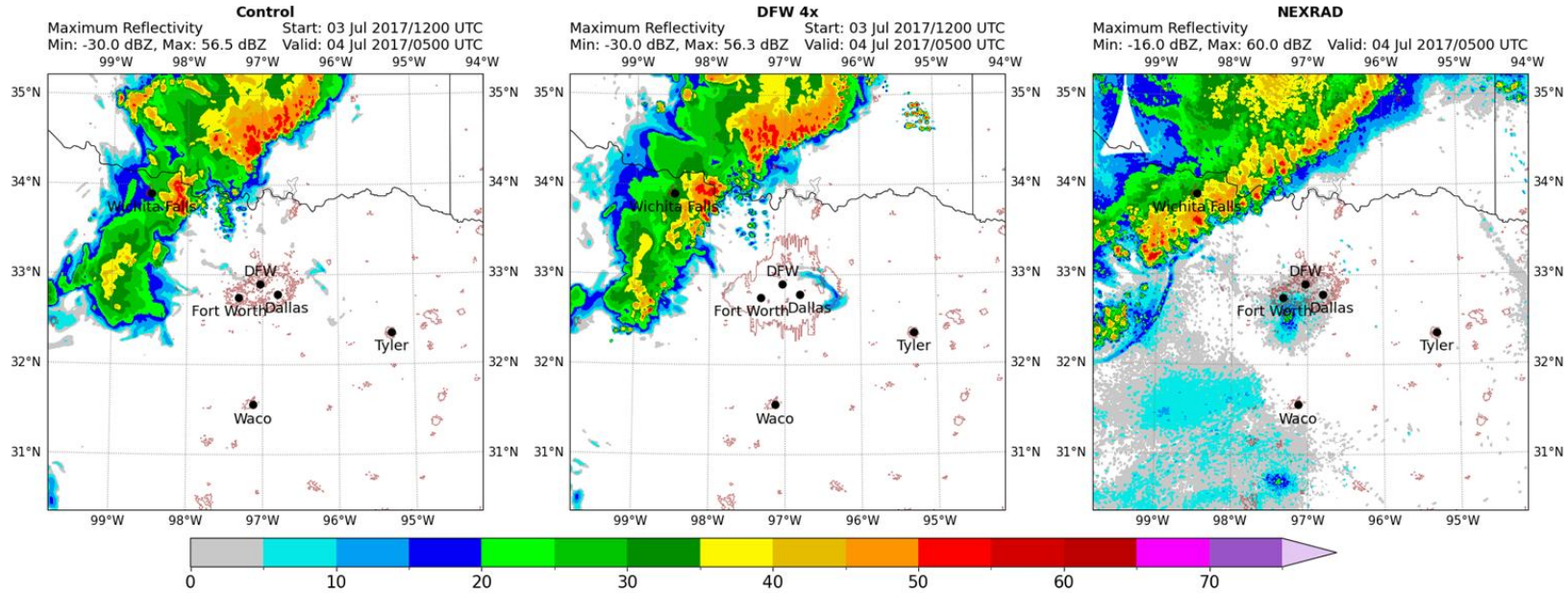


# I-WRF Hurricane Matthew METplus Visualization



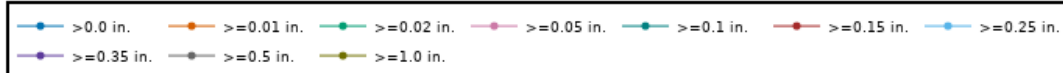
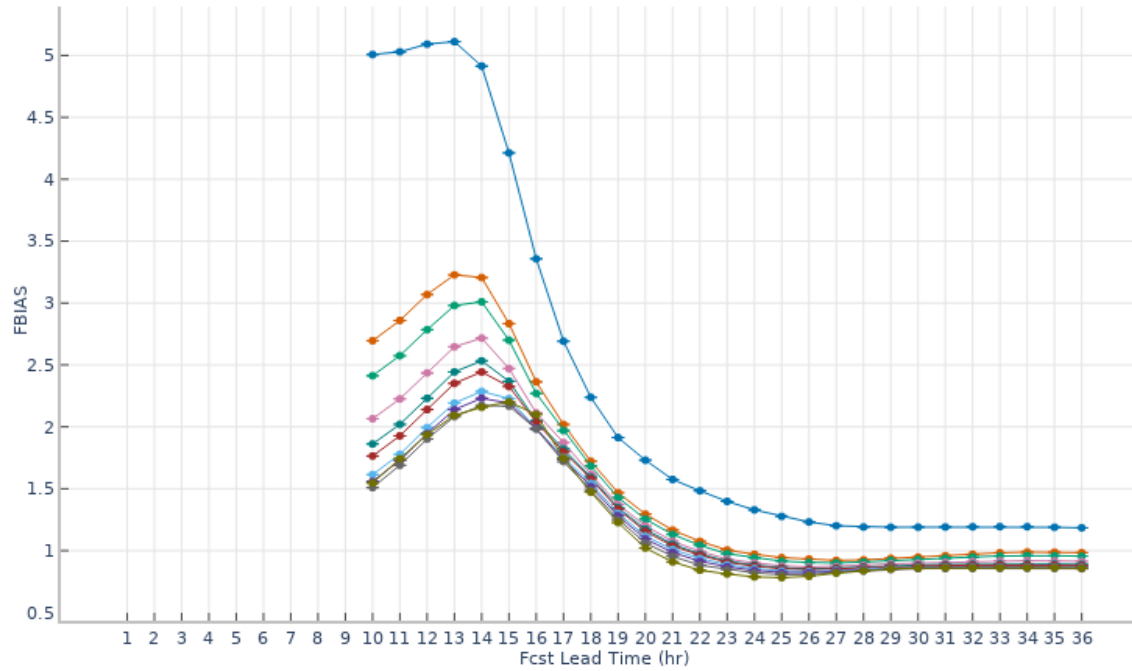
# I-WRF LULC Test Case Python Visualization

WRF Domain 3 — Urban Land Use (outlined)

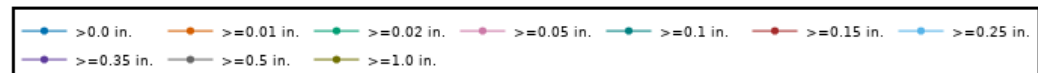
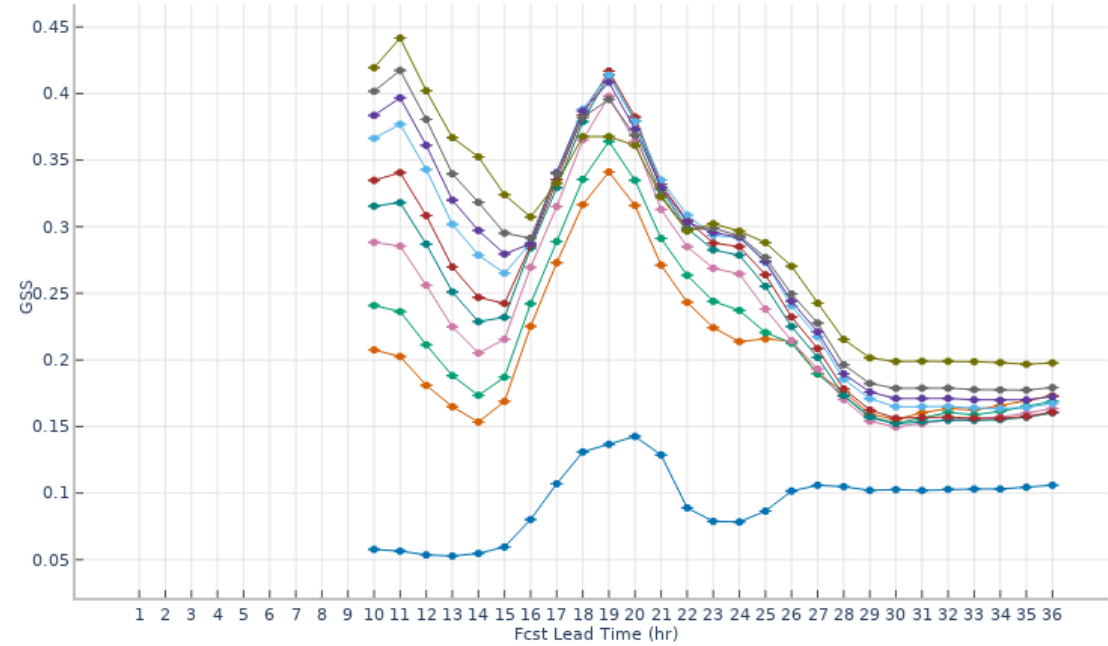


# I-WRF LULC Test Case Python Visualization

## Frequency Bias for 1-Hour Accumulated Precip

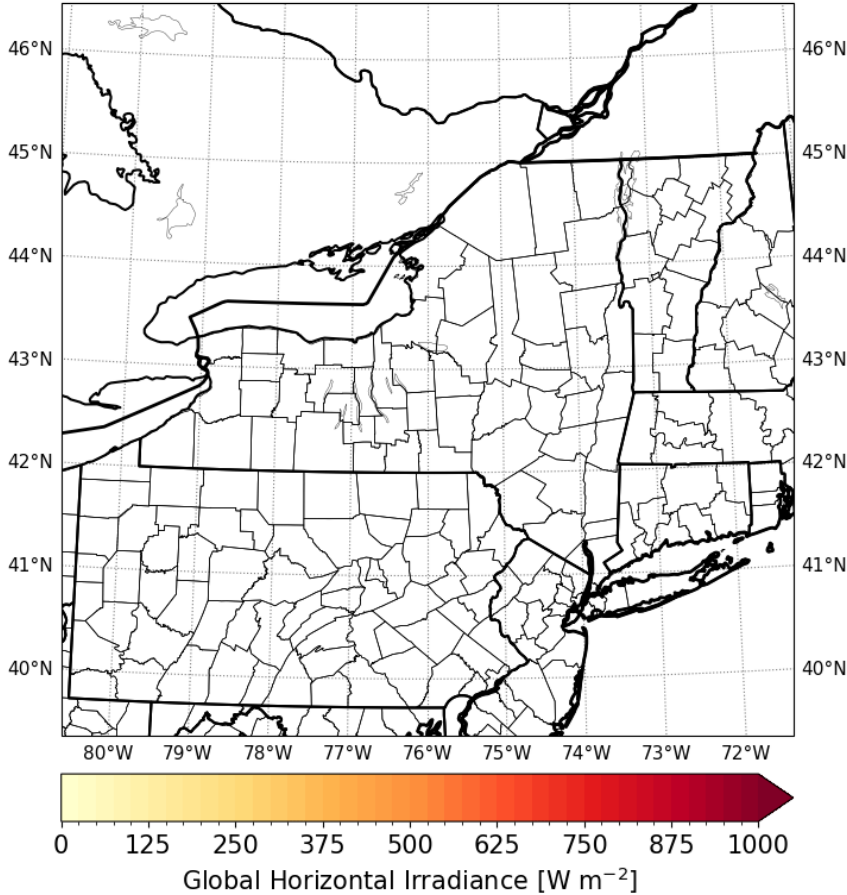


## Gilbert Skill Score for 1-Hour Accumulated Precip

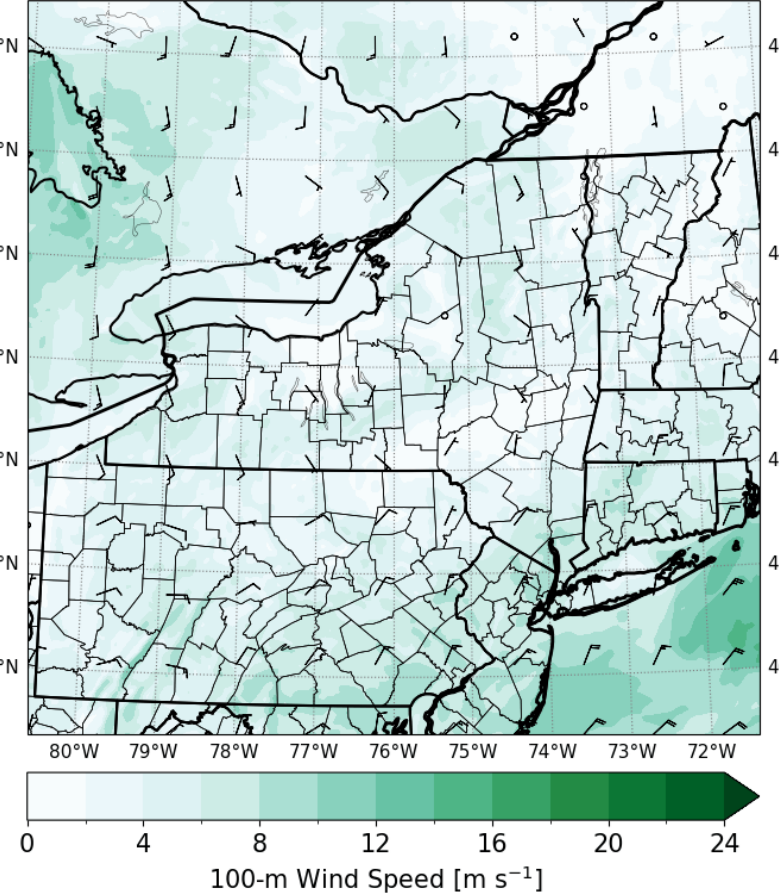


# I-WRF Renewable Energy Test Case Python Visualization

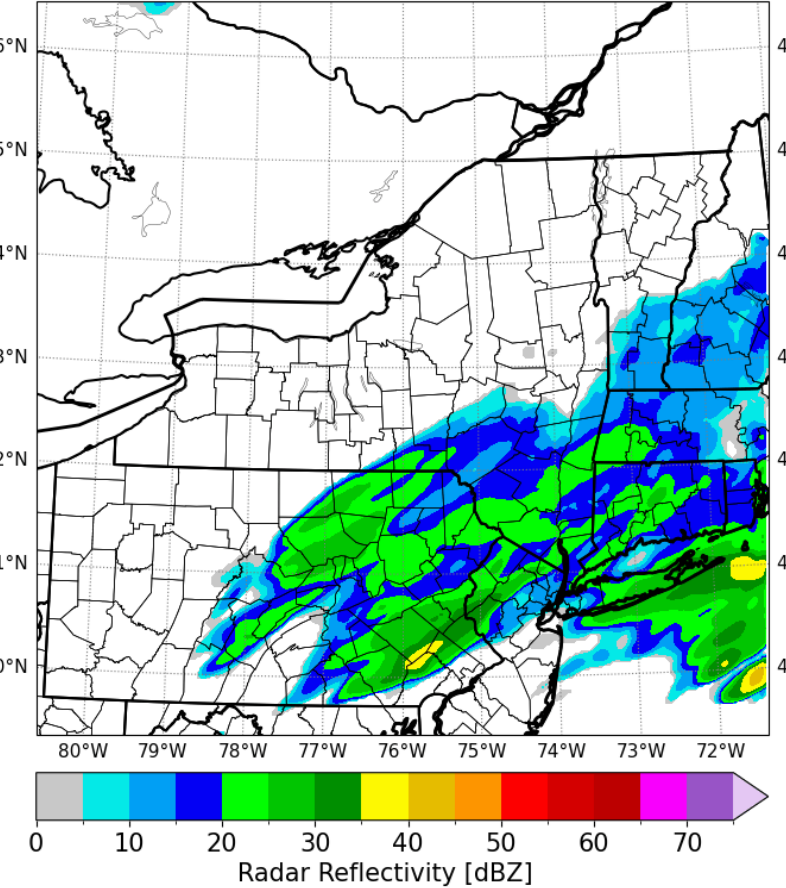
I-WRF Renewable Energy Use Case Demo  
Global Horizontal Irradiance  
Min: 0.0  $\text{W m}^{-2}$ , Max: 0.0  $\text{W m}^{-2}$   
Start: 06 Apr 2025/0600 UTC  
Valid: 07 Apr 2025/1000 UTC  
81°W 80°W 79°W 78°W 77°W 76°W 75°W 74°W 73°W 72°W 71°W



I-WRF Renewable Energy Use Case Demo  
100-m Wind Speed  
Min: 0.0  $\text{m s}^{-1}$ , Max: 14.9  $\text{m s}^{-1}$   
Start: 06 Apr 2025/0600 UTC  
Valid: 07 Apr 2025/1000 UTC  
81°W 80°W 79°W 78°W 77°W 76°W 75°W 74°W 73°W 72°W 71°W

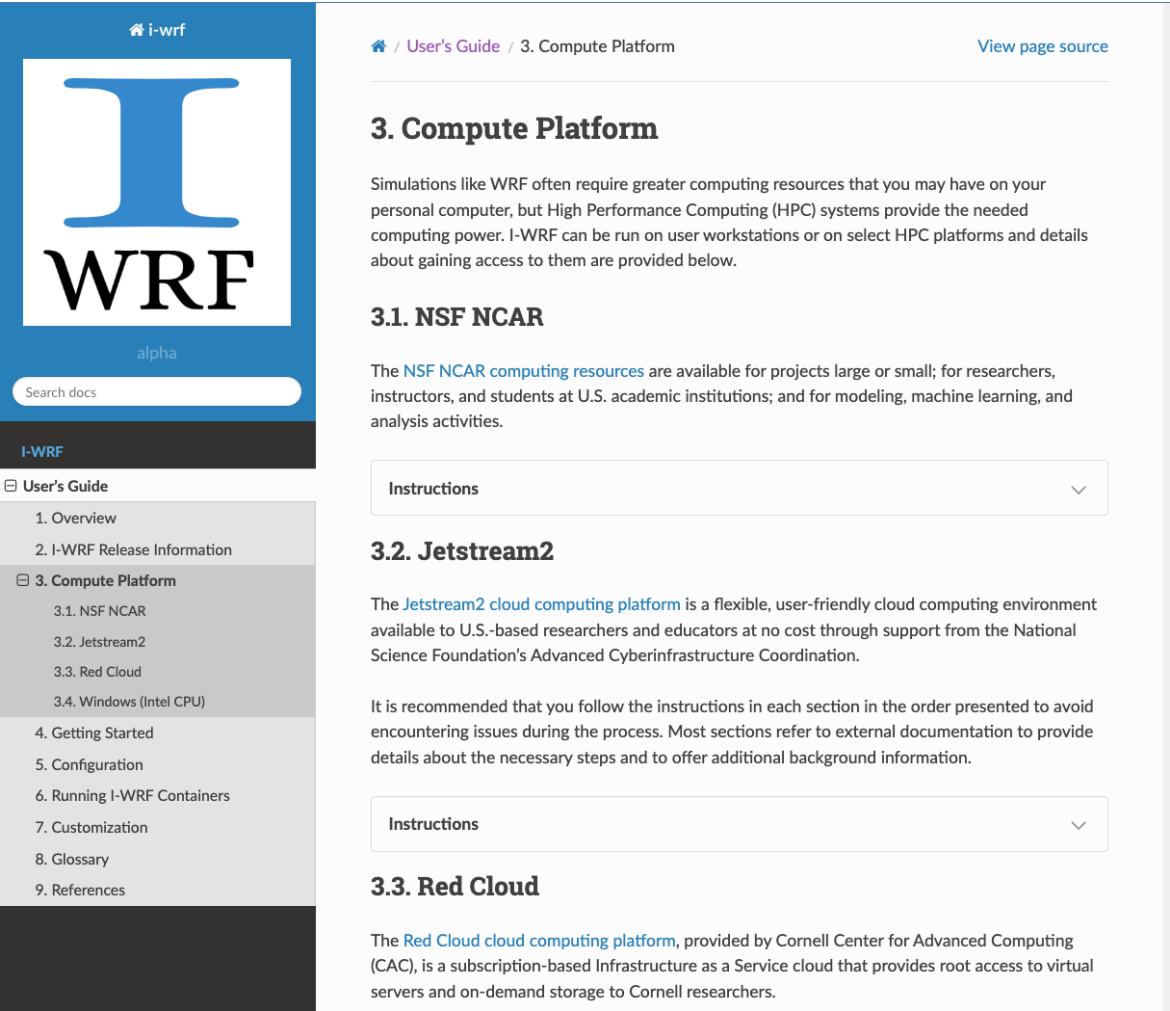


I-WRF Renewable Energy Use Case Demo  
Radar Reflectivity  
Min: -30.0 dBZ, Max: 41.0 dBZ  
Start: 06 Apr 2025/0600 UTC  
Valid: 07 Apr 2025/1000 UTC  
81°W 80°W 79°W 78°W 77°W 76°W 75°W 74°W 73°W 72°W 71°W

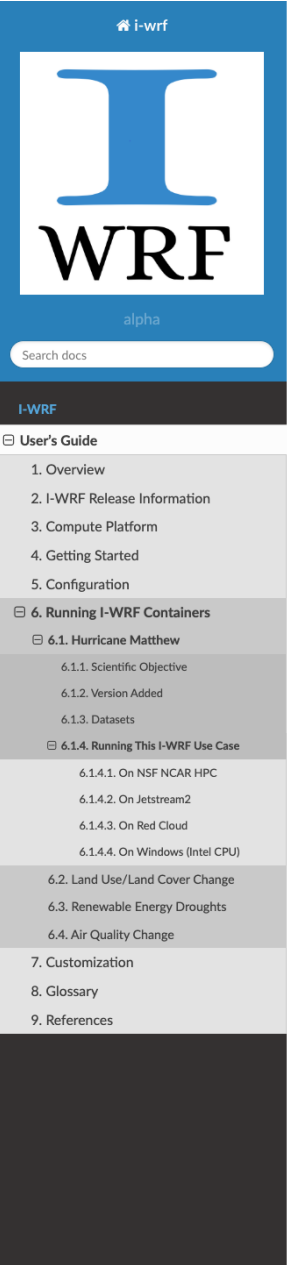


- Observations from New York State Mesonet will be used for verification for this case
  - 127 surface stations
  - 17 profiler stations (with lidars to verify 100-m wind forecasts)

# I-WRF ReadTheDocs Documentation



The screenshot shows the I-WRF ReadTheDocs website. The header includes the I-WRF logo and a search bar. The navigation menu on the left lists sections from 1. Overview to 9. References, with '3. Compute Platform' selected. The main content area is titled '3. Compute Platform' and includes a breadcrumb trail: 'User's Guide / 3. Compute Platform'. Below the title is a paragraph explaining that simulations like WRF require greater computing resources and that I-WRF can run on user workstations or select HPC platforms. The section is divided into three sub-sections: 3.1. NSF NCAR, 3.2. Jetstream2, and 3.3. Red Cloud. Each sub-section has a corresponding 'Instructions' dropdown menu.



The screenshot shows the I-WRF ReadTheDocs website. The header includes the I-WRF logo and a search bar. The navigation menu on the left lists sections from 1. Overview to 9. References, with '6.1.4. Running This I-WRF Use Case' selected. The main content area is titled '6.1.4. Running This I-WRF Use Case' and includes a breadcrumb trail: 'User's Guide / 6.1.4. Running This I-WRF Use Case'. Below the title is a paragraph explaining that with your instance created and running and you logged in to it, you can now install the necessary software and download the data to run the simulation. The section is divided into three sub-sections: 6.1.4.1. On NSF NCAR HPC, 6.1.4.2. On Jetstream2, and 6.1.4.3. On Red Cloud. Each sub-section has a corresponding 'Instructions' dropdown menu.

## 6.1.4. Running This I-WRF Use Case

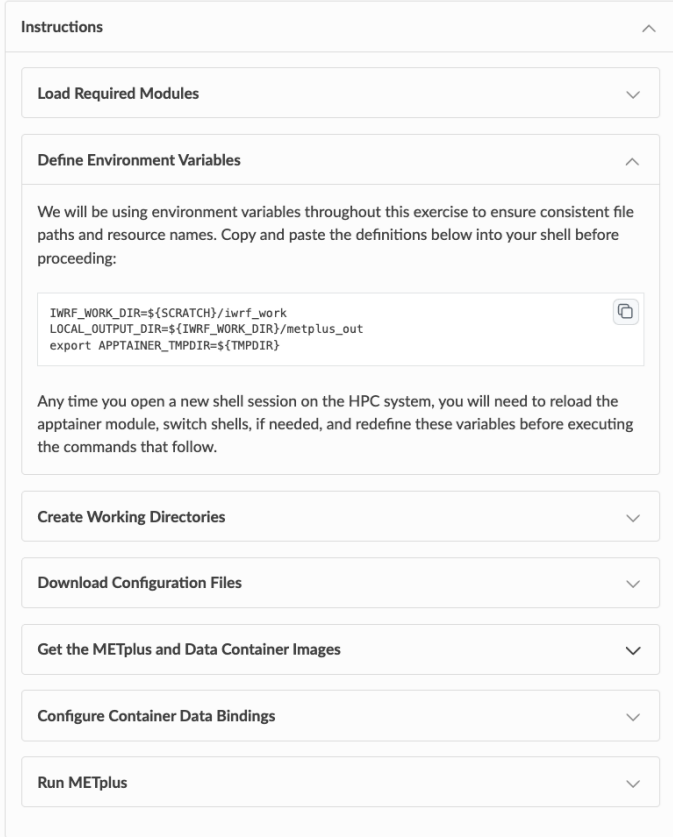
With your instance created and running and you logged in to it, you can now install the necessary software and download the data to run the simulation.

Instructions are provided below for running the Hurricane Matthew use case for each [Compute Platform](#) on which it has been tested.

### 6.1.4.1. On NSF NCAR HPC

Follow the compute platform instructions for [NSF NCAR](#) to secure access to and log in to NSF NCAR HPC.

These instructions are currently limited to running the METplus verification software and assume that WRF output is already available in a local directory.



The screenshot shows the 'Instructions' dropdown menu for the '6.1.4. Running This I-WRF Use Case' section. The menu is titled 'Instructions' and contains several items: 'Load Required Modules', 'Define Environment Variables', 'Create Working Directories', 'Download Configuration Files', 'Get the METplus and Data Container Images', 'Configure Container Data Bindings', and 'Run METplus'. The 'Define Environment Variables' item is expanded, showing a code block with the following content:

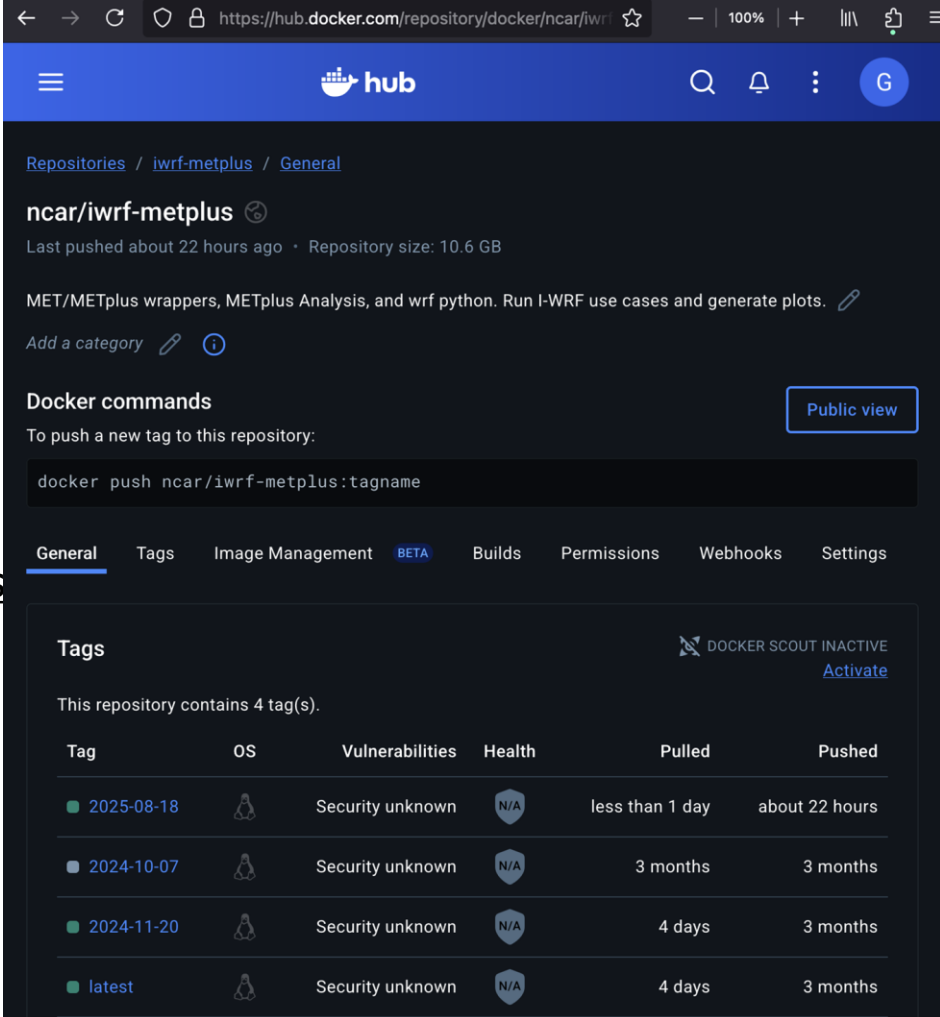
```
IWRF_WORK_DIR=${SCRATCH}/iwrf_work
LOCAL_OUTPUT_DIR=${IWRF_WORK_DIR}/metplus_out
export APPTAINER_TMPDIR=${TMPDIR}
```

### 6.1.4.2. On Jetstream2

- User's Guide: <https://i-wrf.readthedocs.io>

# I-WRF DockerHub Repositories

- **ncar/iwrf** : WRF Container
  - Contains software requires to run WRF
  - <https://hub.docker.com/repository/docker/ncar/iwrf>
- **ncar/iwrf-metplus** : METplus Container
  - Contains METplus wrappers, MET C++ executables, METplus Analysis plotting tools, and WRF-Python package
  - <https://hub.docker.com/repository/docker/ncar/iwrf-metplus>
- **ncar/iwrf-data** : Data Volumes
  - Contains input data used to run I-WRF use cases
  - <https://hub.docker.com/repository/docker/ncar/iwrf-data>



The screenshot displays the DockerHub interface for the repository `ncar/iwrf-metplus`. The page includes a navigation bar with the DockerHub logo and search, notification, and user icons. The repository details section shows the name `ncar/iwrf-metplus`, last pushed time (about 22 hours ago), and repository size (10.6 GB). A description states: "MET/METplus wrappers, METplus Analysis, and wrf python. Run I-WRF use cases and generate plots." Below this, there are options to "Add a category" and a "Public view" button. The "Docker commands" section provides the command: `docker push ncar/iwrf-metplus:tagname`. The "General" tab is selected, showing a "Tags" section with a table of repository tags. A "DOCKER SCOUT INACTIVE" notification is also visible.

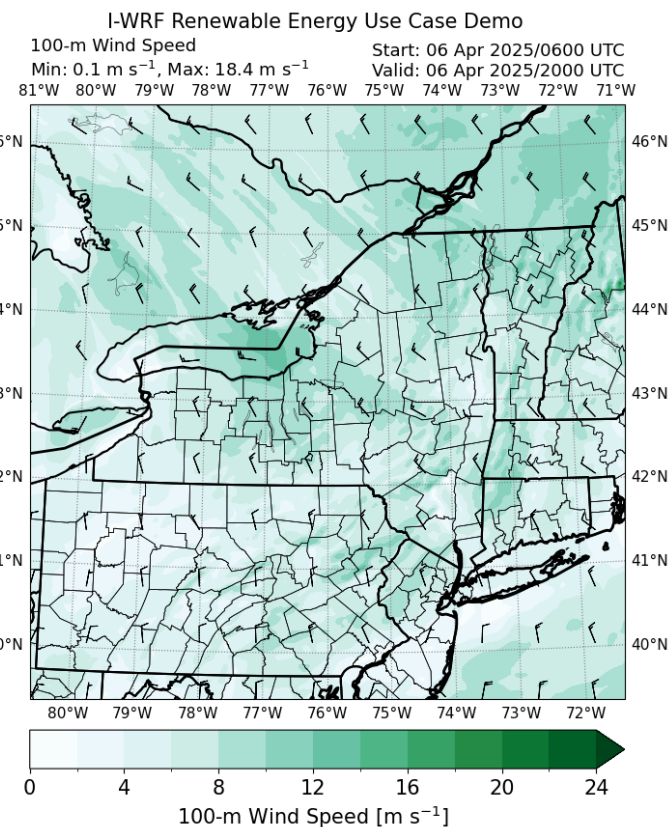
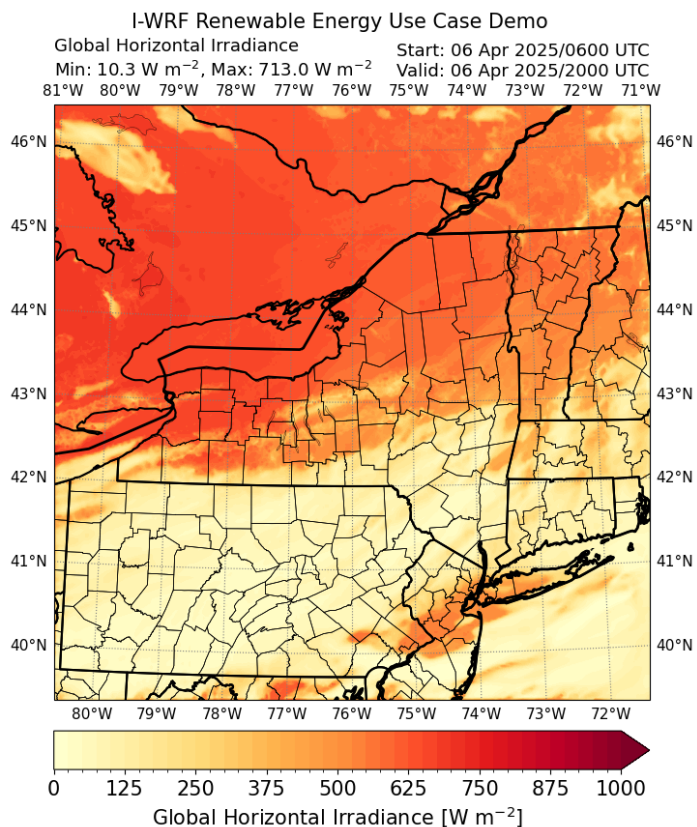
Tag	OS	Vulnerabilities	Health	Pulled	Pushed
2025-08-18	linux/amd64	Security unknown	N/A	less than 1 day	about 22 hours
2024-10-07	linux/amd64	Security unknown	N/A	3 months	3 months
2024-11-20	linux/amd64	Security unknown	N/A	4 days	3 months
latest	linux/amd64	Security unknown	N/A	4 days	3 months

# Major Technical Milestones Accomplished

- I-WRF v0.1
  - Hurricane Matthew demo with WRF & METplus containers (Oct 2024)
- I-WRF v0.2
  - Hurricane Matthew demo with visualization scripts added to METplus container (Dec 2024)
- I-WRF v0.3
  - LULC demo (Aug 2025, formally tagged as a release in Dec 2025)
  - Additional tasks in progress (v0.3.1):
    - Filling out METplus container instructions for Jetstream2, Red Cloud
    - Adding WRF visualization script to METplus container

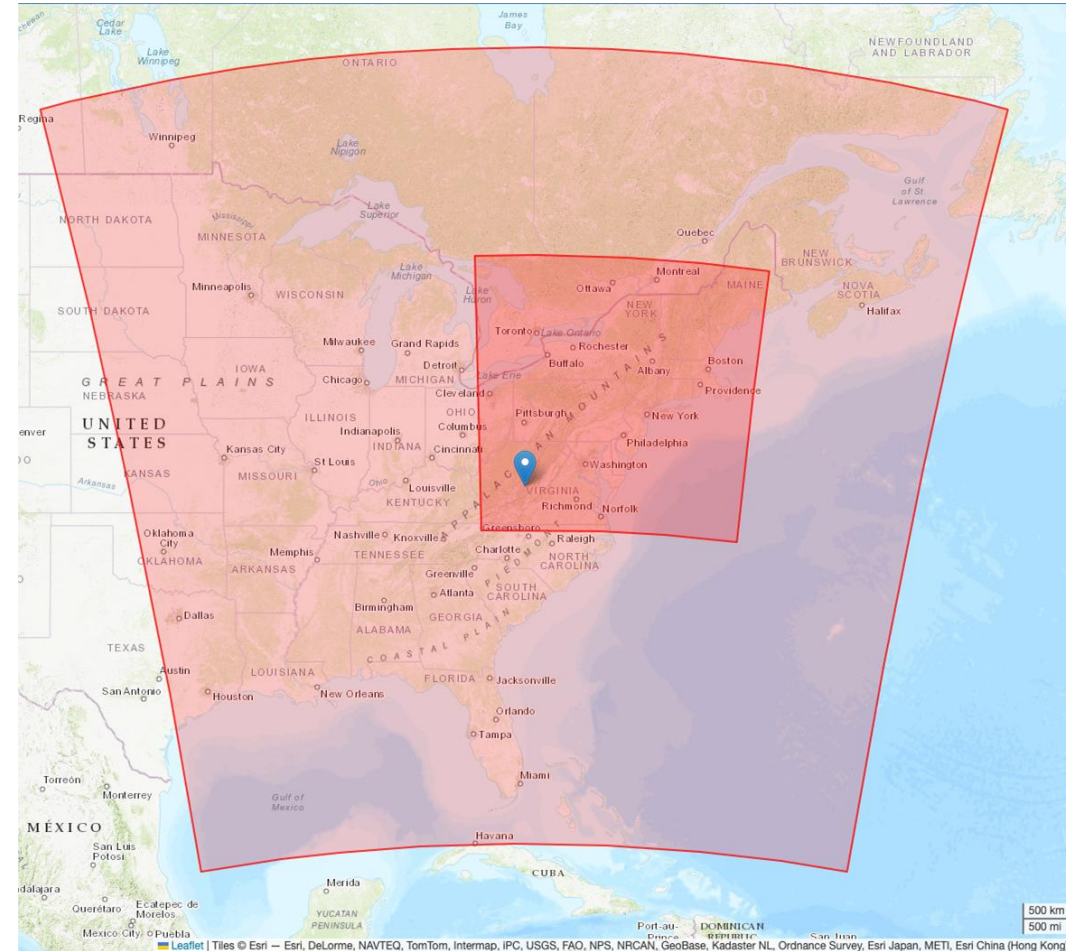
# Major Technical Milestones In Progress

- I-WRF v0.4
  - Renewable Energy demo (in progress)
    - WRF case date/domain finalized, observations requested from NYS Mesonet
    - Adapting scripts from Hurricane Matthew demo for running WRF container
    - Configuring METplus container to verify WRF against NYS Mesonet obs
    - WRF visualization script for this demo mostly done



# Major Technical Milestones In Progress

- I-WRF v0.5
  - Air quality change demo (in progress)
    - WRF-Chem containerized in Apptainer for Derecho and in Docker for Red Cloud and Jetstream2
    - WRF-Chem domain finalized (12/4-km domains over eastern U.S.)
    - Need to identify/obtain obs for METplus container demo
    - Need to adapt WRF visualization scripts



# Major Technical Milestones In Progress

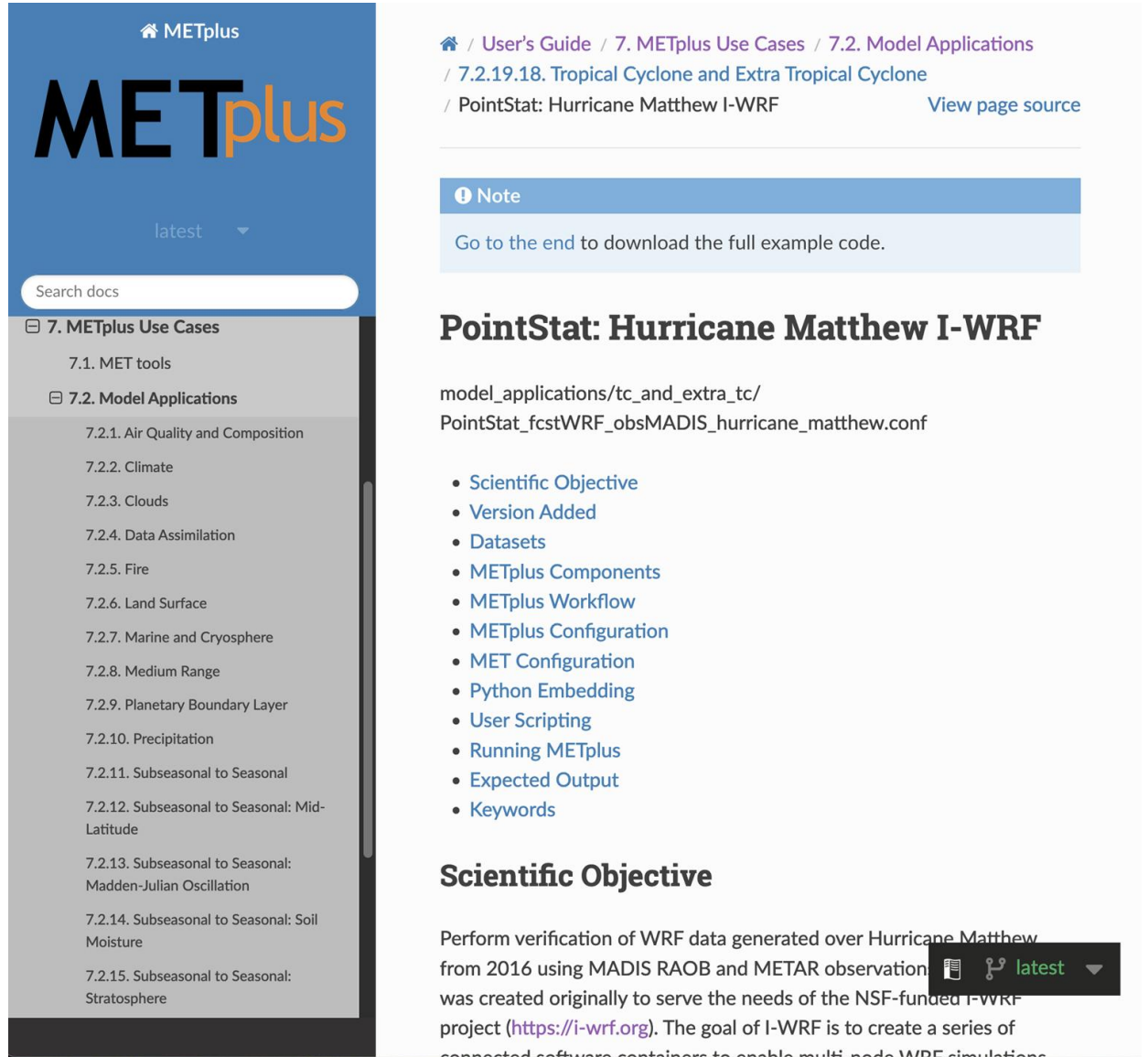
- I-WRF v1.0
  - User customization (in progress)
    - Add user exercises to change WRF physics, METplus stats, visualization
    - Add capability for user to select different date, triggering automatic retrieval of WRF ICs/LBCs and observations for that date
    - Add some level of guidance for users to create their own containers

```
&physics
mp_physics                = 28, 28,
use_aero_icbc              = .true.,
wif_fire_emit             = .false.,
do_radar_ref              = 1,
ra_lw_physics              = 4, 4,
ra_sw_physics              = 4, 4,
radt                      = 10, 10,
swint_opt                 = 2,
slope_rad                 = 1, 1, 1,
topo_shading              = 1, 1, 1,
topo_wind                 = 0, 0, 0,
sf_sfclay_physics         = 5, 5,
sf_surface_physics        = 2, 2,
sf_surface_mosaic         = 1,
mosaic_cat                 = 3,
sf_lake_physics           = 0, 0,
bl_pbl_physics            = 5, 5,
bldt                      = 0, 0,
tke_budget                = 0,
bl_mynn_tkeadvect         = .false.,
bl_mynn_cloudpdf          = 1,
bl_mynn_mixlength         = 2,
bl_mynn_edmf              = 1,
bl_mynn_edmf_mom          = 0,
bl_mynn_edmf_tke          = 0,
grav_settling             = 0,
scalar_pblmix             = 1,
cu_physics                = 0, 0,
cudt                      = 0, 0,
isfflx                    = 1,
ifsnow                    = 1,
icloud                    = 3,
icloud_bl                 = 1,
insert_init_cloud         = .true.,
surface_input_source      = 1,
num_soil_layers           = 8,
sf_urban_physics          = 0,
mp_zero_out               = 2,
mp_zero_out_thresh        = 1.e-12,
use_aero_icbc              = .true.,
```

# **I-WRF GitHub Project Board – Demo**

# METplus Use Case

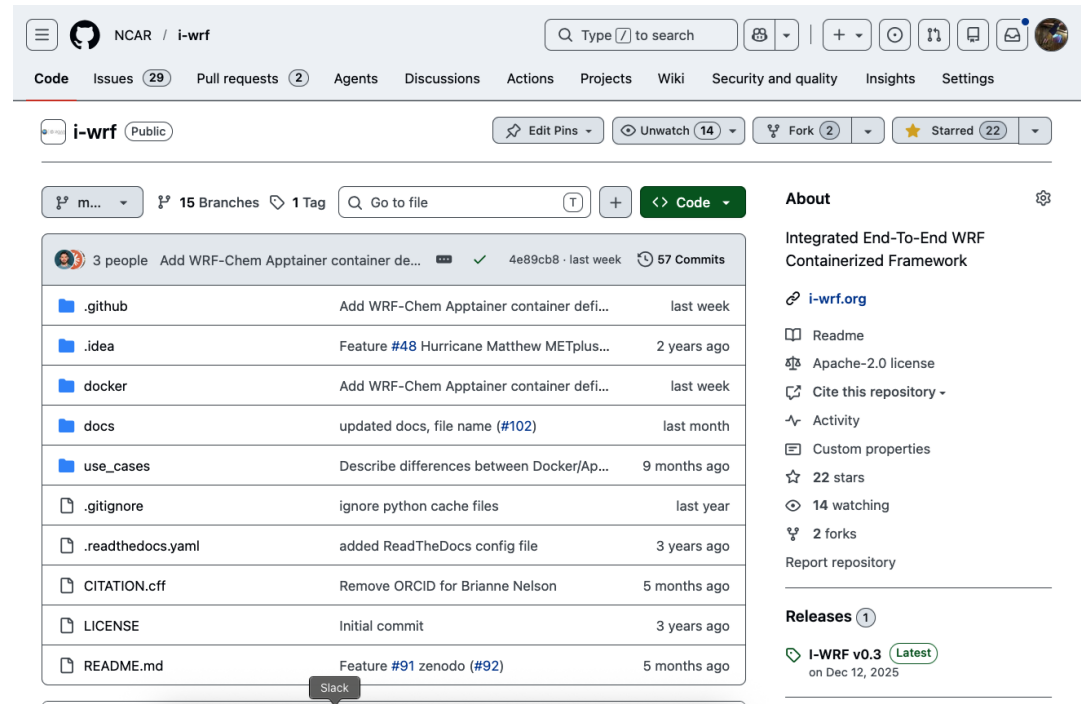
- METplus portion of the I-WRF Hurricane Matthew use case was added to the METplus use cases
- Contributes example of reading WRF into MET that is useful to the community
- Increases visibility of I-WRF project



The screenshot shows the METplus documentation website. The top navigation bar includes the METplus logo and a search bar. The left sidebar contains a table of contents for the '7. METplus Use Cases' section, with '7.2. Model Applications' expanded to show sub-sections like '7.2.1. Air Quality and Composition' through '7.2.15. Subseasonal to Seasonal: Stratosphere'. The main content area displays the breadcrumb path: 'User's Guide / 7. METplus Use Cases / 7.2. Model Applications / 7.2.19.18. Tropical Cyclone and Extra Tropical Cyclone / PointStat: Hurricane Matthew I-WRF'. A 'Note' box contains the text 'Go to the end to download the full example code.' Below this, the title 'PointStat: Hurricane Matthew I-WRF' is followed by the file path 'model\_applications/tc\_and\_extra\_tc/PointStat\_fcstWRF\_obsMADIS\_hurricane\_matthew.conf'. A list of links includes 'Scientific Objective', 'Version Added', 'Datasets', 'METplus Components', 'METplus Workflow', 'METplus Configuration', 'MET Configuration', 'Python Embedding', 'User Scripting', 'Running METplus', 'Expected Output', and 'Keywords'. The 'Scientific Objective' section begins with the text 'Perform verification of WRF data generated over Hurricane Matthew from 2016 using MADIS RAOB and METAR observation...'. A 'latest' dropdown menu is visible in the bottom right corner of the page.

# I-WRF Useful Links

- Run it yourself
  - Hurricane Matthew demo: [https://i-wrf.readthedocs.io/en/latest/Users\\_Guide/use-cases/matthew.html](https://i-wrf.readthedocs.io/en/latest/Users_Guide/use-cases/matthew.html)
  - LULC demo: [https://i-wrf.readthedocs.io/en/latest/Users\\_Guide/use-cases/lulc.html](https://i-wrf.readthedocs.io/en/latest/Users_Guide/use-cases/lulc.html)
- Overview website: <https://i-wrf.org>
- User's Guide: <https://i-wrf.readthedocs.io>
- Github: <https://github.com/NCAR/i-wrf>



*Github repository*

Questions?

Supplemental material

# I-WRF GitHub Project Board – All Deliverables View

- Created an issue for each project deliverable
- Applied label `type: deliverable` to each issue
- Filtered issues with label `type: deliverable`

The screenshot shows the GitHub Project Board for the 'I-WRF Development' project. The board is filtered to show issues with the label 'type: deliverable'. The issues are listed in a table with columns for Title, Status, and Labels.

Title	Status	Labels
1 1.1 Generate desired configuration #10	In progress	deliverable 1.1 type: deliverable
2 1.1.1 Gather requirements #11	In progress	deliverable 1.1 type: deliverable
3 1.1.2 Define base configuration #12	Done	deliverable 1.1 type: deliverable
4 1.1.3 Quality attributes #13	Done	deliverable 1.1 type: deliverable
5 1.2.1 Generate containers: WRF and METplus compilation #14	Done	component: Aptainer component: Docker
6 1.2.2 Generate containers: Visualization component compila... #29	Done	component: plotting deliverable 1.2 type: deliverable
7 1.2.3 Terraform/Kubernetes orchestration #15	Pending	deliverable 1.2 type: deliverable
8 1.2.4 Container refinement #16	Ready	component: Aptainer component: Docker
9 1.2.5 WRF-Chem container #30	Ready	component: Aptainer component: Docker
10 2.1 Use cases: Land Use Land Cover (LULC) #31	Done	deliverable 2.1 type: deliverable
11 2.1.1 Sensitivity to climate change #17	Done	deliverable 2.1 type: deliverable
12 2.1.2 LULC impact on morphology of extreme events #18	Done	deliverable 2.1 type: deliverable
13 2.1.3 Degree of change of societal vulnerability #19	Done	deliverable 2.1 type: deliverable

At the bottom of the board, there is a prompt: '+ You can use `Control + Space` to add an item'.

# I-WRF GitHub Project Board – X.Y Deliverable View

- Created sub-issues for specific, assignable tasks
- Applied **deliverable: X.Y** label to related issues
- Quick view of status, assignee(s), relevant pull requests, labels, and other information

The screenshot shows a GitHub Project Board for the 'I-WRF Development' project. The board is filtered to show issues with the 'deliverable: 1.2' label. The issues are listed in a table with columns for Title, Status, Assignees, Link, and Labels.

Title	Status	Assignees	Link...	Labels
1.2.1 Generate containers: WRF and METplus co... #14	Done			component: Apptainer
Determine where to store Docker images #7	Done	georgemccabe		deliverable 1.2 type: t
Pass native WRF output to MET tools #5	Done	georgemccabe		deliverable 1.2 type: t
Build/run WRF in container using intel compiler #40	Done	hahnd	#72	component: Apptainer
Update container to use Intel OneAPI when WRF ... #45	Done	hahnd		component: Docker
Run/test METplus container using output from W... #41	Done	briannen	#62	deliverable 1.2 type: t
Run Hurricane Matthew test case in WRF contai... #46	Done	rcplane	#62	component: Apptainer
Resolve issues with permissions for files outside o... #4	Done			component: Docker
Update METplus use case to use MADIS2NC wra... #51	Done	georgemccabe		component: METplus
Obtain observation data for Hurricane Matthew t... #47	Done	jaredalee		component: METplus
Run Hurricane Matthew test case in METplus co... #48	Done	georgemccabe...	#53	component: METplus
Run Hurricane Matthew case end-to-end in WRF... #49	Done	georgemccabe		component: METplus
1.2.2 Generate containers: Visualization compon... #29	Done	georgemccabe	#79	component: plotting

# I-WRF GitHub Project Board – Deliverable Issue

## 1.2.1 Deliverable Issue:

- Recently, GitHub added the ability to define issue “relationship” to officially relate parent and sub-issues
- Sub-issues can now be viewed from the parent issue

The screenshot shows a GitHub issue page for the project 'I-WRF Development'. The issue title is '1.2.1 Generate containers: WRF and METplus compilation #14'. It is marked as 'Closed' and has 11 sub-issues. The issue was opened on May 23, 2023, by georgemccabe. The description reads: 'Create containers to run WRF and METplus. Document instructions to run basic use case.' The issue is assigned to no one and has labels: 'component: Apptainer', 'component: Docker', 'deliverable 1.2', and 'type: deliverable'. The sub-issues list includes: 'Determine where to store Docker images #7', 'Pass native WRF output to MET tools #5', 'Build/run WRF in container using intel compiler #40', 'Update container to use Intel OneAPI when WRF 4.5.2 is available', 'Run/test METplus container using output from WRF', 'Run Hurricane Matthew test case in WRF container', 'Resolve issues with permissions for files outside of container', and 'Update METplus use case to use MADIS2NC wrapper when...'. The right sidebar shows the issue is part of the 'I-WRF Development' project, which is marked as 'Done', and is set to 'X-Large' size with a cycle from '2023 - Year 1' (Aug 1 - Jul 31).

# I-WRF GitHub Project Board – Sub-Issue

- Issue details are provided using an issue template
- Add comments to provide updates and ask questions
- Can include links to other I-WRF issues or issues from external repositories
- Status of external issue can be easily viewed from I-WRF issue by mousing over link

The screenshot shows a GitHub issue page for the repository 'NCAR / i-wrf'. The issue title is 'Update METplus use case to use MADIS2NC when it is available #51'. The issue is marked as 'Closed' and has a parent issue '1.2.1 Generate containers: WRF and METplus compilation'. A comment by user 'georgemccabe' is visible, dated April 12, 2024. The comment text reads: 'The MADIS2NC wrapper has not yet been created (issue dtcenter/METplus#1514). The METplus use case (#48) can still call the madis2nc MET tool by using the UserScript wrapper.' Below the comment, there are sections for 'Describe the Task', 'Time Estimate' (~1 day), and 'Sub-Issues' (Consider breaking the task down into sub-issues). A checkbox is checked for 'Add a checkbox for each sub-issue here.' A tooltip is visible over the link 'dtcenter/METplus#1514', showing details for issue 'New Wrapper: MADIS2NC #1514'. The tooltip includes the description 'Describe the New Feature', the text 'Create METplus wrapper to easily configure madis2nc MET tool.', and labels: 'component: python...', 'priority: high', and 'type: new feature'. On the right side, the 'Labels' section shows 'component: METplus', 'deliverable 1.2', 'priority: medium', and 'requestor: NCAR/RAL'. The 'Type' is 'No type'. The 'Projects' section shows 'I-WRF Development' with a status of 'Done'. The 'Milestone' section shows 'I-WRF 1.0.0' with a status of 'Past due by 10m 23d, 100% complete'.

# METplus Development

- MET was enhanced to support reading WRF output files directly without using a Python script
- An issue in the MET repository was created to track this work

The screenshot shows a GitHub issue page for the repository 'dtcenter / MET'. The issue title is 'Add support for native WRF output files already on pressure levels #2547'. The issue is marked as 'Closed' and has 2790 views. It was opened by JohnHalleyGotway on May 17, 2023. The issue description asks for enhancements to MET to read NetCDF WRF output files on pressure levels. It includes sections for 'Describe the New Feature' and 'Acceptance Testing'. The right sidebar shows assignees (JohnHalleyGotway, georgemccabe), labels (MET: Library Code, priority: high, requestor: NCAR/RAL, required: FOR DEVELOPMENT RELE..., type: new feature), and projects (MET-12.0.0 Development).

https://github.com/dtcenter/MET/issues/2547

dtcenter / MET

Code Issues 323 Pull requests Actions Projects 11 Wiki Settings

Add support for native WRF output files already on pressure levels #2547

Edit New issue

Closed #2790

JohnHalleyGotway opened on May 17, 2023 Member

### Describe the New Feature

For the NCAR/RAL I-WRF project, MET needs to be enhanced to directly read NetCDF WRF output files that are already on pressure levels. The breadth of the changes needed are not entirely clear at this time, but MET should be able to interrogate the input NetCDF files to automatically determine whether or not they satisfy the requirement of being reported on pressure levels. Another key question is how to handle the location of velocity points (i.e. winds) which are staggered one half grid points from the mass points. Should that data be processed on the velocity points as-is or automatically interpolated to the mass points?

Also consider automated derivation of wind speed. Is this needed?

Should this be a new file type handle this or should the existing file\_type = NETCDF\_PINT be expanded to include it? Perhaps add NETCDF\_WRF?

### Acceptance Testing

List input data types and sources.  
Describe tests required for new functionality.

Assignees: JohnHalleyGotway, georgemccabe

Labels: MET: Library Code, priority: high, requestor: NCAR/RAL, required: FOR DEVELOPMENT RELE..., type: new feature

Type: No type

Projects: 1 closed project ^ MET-12.0.0 Development Status: Done