EPIC Infrastructure update

Collaborative effort between - Community Collaborators slide attached

Special Acknowledgments: Dr. Mark Potts Dr. Stylianos Flampouris, Dr. Jong Kim, Kris Booker
Agenda

- Partners/EPIC Progress
- We hear you, We want to hear more
- Community Infrastructure
  - Repeatable to SRW, RRFS, LandDA, HAFS, and future applications
- CI/CD
  - Complete for SRW
  - Repeatable for LandDA, HAFS, and future applications
- Tutorials and training
- Closing
- Need for Testing
- Closing
Partners
Community Collaborators/Partners

Acknowledgement

- NOAA OAR: WPO, GSL, PSL, NSSL, CSL, AOML, GFDL
- NOAA Open Data Dissemination (NODD) Program
- NWS: EMC, OSTI
- DTC
- UCAR: CGD, JCSDA
- Academia: George Mason University, Oklahoma University, University of Michigan
- CSPs: AWS, Azure, and Google Cloud
- Cooperative Institutes: CIRES, CIMSS
EPIC Progress
We hear you, we want to hear more!

- We need fewer users and more contributors.
- How do I replicate EPIC event infrastructure?
- How can we track contributors versus users across applications?
- How can we utilize repeatable processes (CI/CD) to test applications have passed all gates?
- Peer reviews need to be faster.
- More tutorials:
  - Contributing to UFS GitHub
  - GitHub Discussion and how to get user support
  - Azure AZ-HOP
- Don’t wait for a survey/meeting email: support.epic@noaa.gov
Simplifying NOAA’s Operational Forecast Suite
Transitioning 21 of NOAA’s Operational Forecast Systems into Eight Applications

Uccellini et al. (BAMS, 2022)
EPIC Architectural Plans

Continued Plans:
- CI/CD across more applications
- Transparent Gates
- Fail or Succeed Quickly
- Enhanced testing frameworks
- Advanced User Support
- Configuration Management
- Cloud configuration scripts
- Community Tools
- Unified Workflow
- Community Events
Community Infrastructure
## EPIC Cloud Architecture

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AWS RTX Sandbox</strong></td>
<td>An AWS account that sits outside of the NOAA firewall to allow for non-CAC users.</td>
</tr>
<tr>
<td><strong>AWS ACIO Sandbox</strong></td>
<td>An AWS account that sits outside of the NOAA firewall to allow for non-CAC users. Login.gov access.</td>
</tr>
<tr>
<td><strong>Azure RTX and ACIO Sandbox</strong></td>
<td>An Azure account that sits outside of the NOAA firewall to allow for non-CAC users.</td>
</tr>
<tr>
<td><strong>GCP ACIO Sandbox</strong></td>
<td>A GCP account that sits outside of the NOAA firewall to allow for non-CAC users.</td>
</tr>
<tr>
<td><strong>AWS ACIO Dev</strong></td>
<td>An AWS account that sits inside the NOAA firewall containing our application code that is in active development under ACIO.</td>
</tr>
<tr>
<td><strong>AWS ACIO Prod</strong></td>
<td>An AWS account that sits inside the NOAA firewall containing our application code that is live in the production account under ACIO.</td>
</tr>
<tr>
<td><strong>Parallel Works</strong></td>
<td>A third-party HPC provider inside of the NOAA firewall that provides us virtual machines to test UFS applications on all 3 cloud service providers.</td>
</tr>
</tbody>
</table>
New Tutorial - Infrastructure as Code

- Common Infrastructure using Packer - able to deploy on any CSP
  - [https://github.com/NOAA-EPIC/packer-srwcluster](https://github.com/NOAA-EPIC/packer-srwcluster)
    - 12 lines of code to build out and run SRW
    - Tutorials - Earth Prediction Innovation Center (noaa.gov)

- Video 2: Creating a Head Node in AWS
  - Can be any of the CSP's

- Video 3: Running any application
  - Starting with SRW v2.1
  - Next: LandDA
CI/CD Pipeline
**Pipeline Gates**

- Average Build time
- Average time per gate
- Average build time per platform
- Code Coverage
- Forecast Skill

<table>
<thead>
<tr>
<th>Pipeline Gate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Checkout Source Code</strong></td>
<td>Pull source code from GitHub and stage the data for analysis before deploying the code.</td>
</tr>
<tr>
<td><strong>Unit Testing</strong></td>
<td>Run available unit tests for projects and ensure that the tests run as expected. Collect code coverage metrics for the available baselines.</td>
</tr>
<tr>
<td><strong>Lint (Flake 8)</strong></td>
<td>Perform static code analysis that enforces style consistencies across programming languages.</td>
</tr>
<tr>
<td><strong>Dependency Check</strong></td>
<td>Scan third-party libraries and modules for current vulnerabilities.</td>
</tr>
<tr>
<td><strong>Build the Cloud Stack</strong></td>
<td>Terraform/Cloudformation scripts will create a repeatable process for deploying applications.</td>
</tr>
<tr>
<td><strong>Lint Cloud Stack</strong></td>
<td>Examine the cloud stack template and return various suggestions.</td>
</tr>
<tr>
<td><strong>Nag Cloud Stack</strong></td>
<td>Pinpoint security vulnerabilities in cloud stack templates.</td>
</tr>
<tr>
<td><strong>Scan Secrets</strong></td>
<td>Scan for any improper use of security passwords or credentials.</td>
</tr>
<tr>
<td><strong>Static Code Analysis</strong></td>
<td>Scan code in all programming languages using SonarQube to determine current vulnerabilities, maintenance issues, and defects. Note: SonarQube also has the ability to utilize architectural metrics such as cyclomatic complexity and maintainability metrics. Cyclomatic complexity as the example infers is a value that tells the ability that a new engineer will be able to come in and maintain the baseline code. If the number is high, then you have an application that is tough to upkeep, so tracking this number over time will make sure that your application is easy to maintain, which in turn reduces technical debt costs.</td>
</tr>
<tr>
<td><strong>Package/Pull Artifacts/Deploy</strong></td>
<td>This gate sequence will package up the artifacts and the application and deploy the application as needed after completing all quality gate checks.</td>
</tr>
<tr>
<td><strong>Run Regression Tests</strong></td>
<td>Run a list of regression tests to test the overall end-to-end functionality.</td>
</tr>
</tbody>
</table>
CI/CD Pipeline

- Master Pipeline:
### Selenium Test Results
Latest Run: 2023-06-12T12:58:26

<table>
<thead>
<tr>
<th>Title Response</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Found Earth Prediction Innovation Center - Site for EPIC, the Earth Prediction Innovation Center.</td>
<td>Success</td>
</tr>
<tr>
<td>Found Short Range Weather Application - Earth Prediction Innovation Center.</td>
<td>Success</td>
</tr>
<tr>
<td>Found Medium Range Weather Application - Earth Prediction Innovation Center.</td>
<td>Success</td>
</tr>
<tr>
<td>Found UFS Weather Model - Earth Prediction Innovation Center.</td>
<td>Success</td>
</tr>
<tr>
<td>Found Land Data Assimilation (DA) System - Earth Prediction Innovation Center.</td>
<td>Success</td>
</tr>
<tr>
<td>Found Unified Post Processor - Earth Prediction Innovation Center.</td>
<td>Success</td>
</tr>
<tr>
<td>Found Get Support - Earth Prediction Innovation Center.</td>
<td>Success</td>
</tr>
<tr>
<td>Found Getting Started - Earth Prediction Innovation Center.</td>
<td>Success</td>
</tr>
<tr>
<td>Found Tutorials - Earth Prediction Innovation Center.</td>
<td>Success</td>
</tr>
<tr>
<td>Found Technical FAQs - Earth Prediction Innovation Center.</td>
<td>Success</td>
</tr>
<tr>
<td>Found News - Earth Prediction Innovation Center.</td>
<td>Success</td>
</tr>
</tbody>
</table>

Overall EPIC Selenium Test Status: Success
## EPIC Dashboard - GitHub Traffic

### UFS_UTILS

UFS_UTILS Github repository

Data from: 2023-05-21 to 2023-06-11

<table>
<thead>
<tr>
<th>Name</th>
<th>Email</th>
<th>Commits</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeorgeG@yahoo-NOAA</td>
<td>53799452+GeorgeG@<a href="mailto:yahoo-NOAA@users.noreply.github.com">yahoo-NOAA@users.noreply.github.com</a></td>
<td>1</td>
</tr>
</tbody>
</table>

### ufs-weather-model

ufs-weather-model Github repository

Data from: 2023-05-21 to 2023-06-11

<table>
<thead>
<tr>
<th>Name</th>
<th>Email</th>
<th>Commits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sadeghi Sadeghi Tabae</td>
<td><a href="mailto:31417685+SadeghiTabae-NOAA@users.noreply.github.com">31417685+SadeghiTabae-NOAA@users.noreply.github.com</a></td>
<td>1</td>
</tr>
<tr>
<td>jamie.wang</td>
<td><a href="mailto:jamie.wang@noaa.gov">jamie.wang@noaa.gov</a></td>
<td>1</td>
</tr>
<tr>
<td>Gillian Petro</td>
<td><a href="mailto:96088083+gipetro-NOAA@users.noreply.github.com">96088083+gipetro-NOAA@users.noreply.github.com</a></td>
<td>1</td>
</tr>
<tr>
<td>Dustin Swales</td>
<td><a href="mailto:dustin.swales@noaa.gov">dustin.swales@noaa.gov</a></td>
<td>1</td>
</tr>
<tr>
<td>dikokron</td>
<td><a href="mailto:dikokron@users.noreply.github.com">dikokron@users.noreply.github.com</a></td>
<td>1</td>
</tr>
</tbody>
</table>
# EPIC Dashboard - Pipeline

**EPIC CI Build Status** - ufs-srweather-app

Last updated: Sun Mar 12 22:42:01 PDT 2023

<table>
<thead>
<tr>
<th>timestamp</th>
<th>PR-build</th>
<th>inProgress</th>
<th>duration (min)</th>
<th>result</th>
<th>WE2E-tests</th>
<th>S3-artifacts</th>
</tr>
</thead>
<tbody>
<tr>
<td>2023-03-10 15:29:36</td>
<td>ufs-srweather-app/job pipeline/job/PR-667:1</td>
<td>true</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-03-10 17:13:46</td>
<td>ufs-srweather-app/job pipeline/job/PR-663:1</td>
<td>false</td>
<td>309.8</td>
<td>FAILURE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-03-08 17:06:26</td>
<td>ufs-srweather-app/job pipeline/job/PR-657:3</td>
<td>false</td>
<td>518.3</td>
<td>FAILURE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-03-08 17:00:05</td>
<td>ufs-srweather-app/job pipeline/job/PR-657:2</td>
<td>false</td>
<td>1</td>
<td>FAILURE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-03-08 16:53:00</td>
<td>ufs-srweather-app/job pipeline/job/PR-657:1</td>
<td>false</td>
<td>0</td>
<td>FAILURE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-03-10 15:08:26</td>
<td>ufs-srweather-app/job pipeline/job/PR-656:1</td>
<td>false</td>
<td>274.9</td>
<td>SUCCESS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-03-08 19:16:35</td>
<td>ufs-srweather-app/job pipeline/job/PR-650:1</td>
<td>false</td>
<td>438.7</td>
<td>FAILURE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-03-09 01:43:47</td>
<td>ufs-srweather-app/job pipeline/job/PR-637:2</td>
<td>false</td>
<td>117</td>
<td>FAILURE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-03-08 16:29:15</td>
<td>ufs-srweather-app/job pipeline/job/PR-637:1</td>
<td>false</td>
<td>554.5</td>
<td>FAILURE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-03-06 16:44:27</td>
<td>ufs-srweather-app/job pipeline/job/PR-632:1</td>
<td>false</td>
<td>167.3</td>
<td>FAILURE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-02-24 18:37:40</td>
<td>ufs-srweather-app/job pipeline/job/PR-628:1</td>
<td>false</td>
<td>218.5</td>
<td>SUCCESS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-03-03 18:51:37</td>
<td>ufs-srweather-app/job pipeline/job/PR-627:1</td>
<td>false</td>
<td>432.1</td>
<td>FAILURE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-02-23 16:50:45</td>
<td>ufs-srweather-app/job pipeline/job/PR-626:1</td>
<td>false</td>
<td>140.6</td>
<td>ABORTED</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Implemented Process

UFS-SRW Application - Example

- **Infrastructure** – Update the CICD pipeline of the SRW to include the driver for forecast verification
- **Scientific Hypothesis** – Evaluate the impact on severe winter weather with the relevant UFS case to validate the hypothesis, i.e., Indianapolis case
- **Objective Verification** – Calculate skill score index based on weighted average of a combination of metrics (RMSE), variables (wind speed, dew point temperature, temperature, and pressure at the lowest level in the atmosphere), and lead time
- **Output** – Every source code update has a performance indicator; i.e., aiming for higher than 1.0.

Next Steps

- Expansion of the infrastructure to all the UFS repositories, already in the Land DA
- Well-established problems with focused research and development
- Significant increase to HPC resources dedicated to the testing, i.e., currently the SRW testing is not triggered due to the lack of resources
- Support for incremental change in development mindset, there are already great examples
Tutorials and Training
Community Engagement Activities

Community Portal and Resources
- Regular Updates, FAQs
- Detailed descriptions of products and Services
- Feedback Pages / Incorporating Feedback

Social Media Campaigns
- Twitter
- Facebook
- Instagram

Webinars and Workshops
- Host webinars and workshops for EPIC community
- Topics related to EPIC, model dev and data analysis

Community Events
- Application Training
- CodeFest
- UIFCW

Publications and Newsletters
- Publish latest developments
- Articles, impacts and contributions
- Guides and technical documents for users

Outreach and Marketing
- Increase awareness of EPIC and community
- Collaborate with external partners and stakeholders
- Targeted messaging and communications strategies
Upcoming Events, Projects & Promotions

- Quarterly CodeFests & Application Trainings
  - Short-Range Weather CodeFest 2023: Unit Testing Framework for UFS - April 3-7, 2023
  - Short-Range Weather Application Training 2023: Running V2.1 Containers in AWS - April 7, 2023
  - EPIC Application Training June 2023: Land Data Assimilation (DA) System - June 23, 2023

- Plan/Host UIFCW, Summer 2023

- Quarterly Video Tutorials

- UPP webpage on ECP (support transitioned from DTC to EPIC)

- Explore combining EPIC-UFS Communications Strategy

- Conferences (AGU & AMS)

- Launch an EPIC-UFS Newsletter

- Develop an enhanced metrics dashboard for the ECP

- Identify potential areas for improvement and engagement, discover new ways to incentivize external participation
Need for testing and governance
Testing Framework

Objective: To quantify the impact of any code update, in terms of forecast accuracy and computational performance.

- Homogenize testing infrastructure
- Optimized testing (Reduction of cost)
- Simplification of Code Management
- User-friendly
- Multi-level testing

## UIFCW 2023
A UFS Collaboration Powered by EPIC
Infrastructure as an Innovation’s Catalyst

Work in Progress by EPIC contract with the UFS Community

Component Catalysts

- Testing Framework
- Code Management
- CI/CD
- Data Management
- Code Optimization
- Containerization
- Model Performance
- Code Analysis
- Agile Development

UFS Performance Improvement

Creation of a prosperous environment for rapid innovation!
We value continuous feedback

Our Advanced User Support Team is prepared to assist as needed with your technical questions.

Await your transparent tools
- Community Dashboard
- Infrastructure as code
- CI/CD Pipeline results are public for all applications

More tutorials:
- Contributing to UFS GitHub
- GitHub Discussion and how to get user support
- Azure AZ-HOP

Don’t wait for a survey/meeting email: support.epic@noaa.gov