EPIC Application Overview

## How NOAA's Unified Forecast System (UFS) Works

The NOAA Unified Forecast System (UFS) is a community-based, coupled, comprehensive Earth modeling system designed to improve weather forecasting accuracy and accelerate the transition of research into operational use. It integrates various model components, data assimilation techniques, and workflow management systems within a unified framework, allowing for a flexible and adaptable system capable of producing a wide range of forecasts, from short-term weather predictions to longer-term climate outlooks.**Key Operational Principles:**

**1. Community-Based Development:**

The UFS is developed and maintained collaboratively by researchers, developers, and users from NOAA, educational institutions, government agencies, and the private sector. This community approach ensures diverse perspectives and expertise, leading to robust and innovative solutions.

**2. Unified Framework:**

As a modular system, the UFS is built from reusable components that can be combined and configured for various forecasting needs. These components include numerical models (atmospheric, ocean, land surface), data assimilation systems, and post-processing tools. Its flexible design allows for the integration of new research and technologies as they become available.

**3. Core Components:**

* **Numerical Models:** The UFS incorporates a suite of coupled numerical models that simulate the physical processes of the Earth's atmosphere, oceans, land surface, and other components.
* **Data Assimilation:** Advanced data assimilation techniques integrate observations from various sources (satellites, weather stations) to create accurate initial states for the models.
* **Post-processing:** Tools are provided for post-processing model output, including statistical analysis, visualization, and user-friendly forecast creation.
* **Workflow Management:** The system includes workflow management tools to automate the forecasting process, from data ingestion to product delivery.

**4. Diverse Applications:**

The UFS can be configured into various applications tailored to specific forecasting needs, such as:

* **Short-range weather forecasting:** High-resolution forecasts for severe weather events like thunderstorms, hurricanes, and tornadoes.
* **Subseasonal to seasonal forecasting:** Predictions for longer-term weather patterns, such as droughts and heat waves.
* **Climate modeling:** Simulations of long-term climate trends and projections.  
  Each application utilizes different combinations of UFS components to achieve its specific goals.

**5. Research to Operations Transition:**

The UFS accelerates the transition of research advancements into operational weather forecasting. By providing a common platform for research and operations, it facilitates the rapid adoption of new scientific findings and technological innovations, ensuring NOAA's weather forecasts are based on the latest scientific understanding and technological capabilities.

POC Scoping Questions

To successfully scope a Proof of Concept (POC) for a Unified Forecast System (UFS) application running on OpenShift with an AI component, it's crucial to ask detailed questions that align with NOAA EPIC's stated objectives and challenges. Based on the provided sources and our conversation, here's a comprehensive list of scoping requirements and associated questions:

### **1. UFS Application and AI Component Specifics**

* **UFS Application Selection:** Which specific UFS application or component (e.g., Short-Range Weather (SRW) App, Land Data Assimilation (DA), Unified Post Processor (UPP), Stochastic Physics, Global App, Hurricane Analysis and Forecast System (HAFS), Joint Effort for Data Assimilation Integration (JEDI)) is the primary candidate for this POC?
  + *Follow-up:* Can you provide access to the relevant GitHub repositories for the chosen UFS application(s)?
* **Core Dependencies:** What are the current core dependencies of this UFS application, including specific compilers (e.g., Intel LLVM), MPI versions, and middleware libraries (e.g., ESMF, FMS, Spack-stack)?
  + *Follow-up:* How frequently do these libraries and compilers typically receive updates?
* **AI Model Integration:** Which specific AI/ML model or hybrid model is intended for integration in this POC?
  + *Follow-up:* What is the current stage of maturity for this AI/ML model (e.g., research prototype, pre-production, operational)?
  + *Follow-up:* What are the key AI/ML operations needed: model development, training, tuning, serving, monitoring, or a combination?
* **Data Requirements:** What are the typical data sources and formats required for running this UFS application and for training/inferencing the AI component (e.g., UFS/NOAA analyses, ERA5, CONUS404, GDAS, HRRR analyses)?
  + *Follow-up:* How are these datasets currently managed and ingested for model use?

### **2. Addressing Current Operational Challenges**

* **Standardized Development Environment:** How would a "templated development space" with pre-deployed IDEs and necessary components specifically benefit your developers, beyond simply providing consistency?
  + *Follow-up:* What is the current average time spent on configuring components for a new developer or environment setup (e.g., the "month to configure MPI" example)?
* **Containerization Standardisation:** What level of "standardized container framework" (e.g., Docker, Singularity, Podman) do you aim to achieve, and how would this address current "volunteer-based" efforts and informal Docker recipes?
  + *Follow-up:* What are the existing challenges with building and deploying UFS code in containers across different HPC systems and compilers?
* **High-Resolution Workflows:** Can you provide more details on the challenges encountered with deploying "high-resolution global workflows on cloud platforms," specifically the root causes of complexity and debugging costs that led to halting efforts?
  + *Follow-up:* What specific resolution (e.g., "half degree or quarter degree," "1.25, 2.5, 5, 10 kilometer") would be the target for demonstrating high-resolution capabilities in the POC?

### **3. CI/CD and Automation Needs**

* **Current CI/CD Practices:** What CI/CD tools are currently in use (e.g., Jenkins)? How are code changes, pull requests, and regression tests managed in your current process?
* **Automation Scope:** What specific build, test, and deployment steps for the UFS application and AI component do you prioritize for automation in this POC using OpenShift Pipelines (Tekton) and OpenShift GitOps (Argo CD)?
  + *Follow-up:* What level of "Infrastructure-as-Code (IaC)" maturity do you aim to demonstrate for environment setup and application deployment?
* **Testing Frameworks:** How can the POC integrate with or enhance the existing UFS testing framework, including regression tests (RT) and operational requirements tests (ORT)?

### **4. Hybrid Cloud and HPC Integration**

* **Deployment Environment:** Where do you envision this POC running (e.g., existing NOAA HPC, specific cloud provider like AWS/PW Cloud/Azure, or a combination)?
  + *Follow-up:* What are the specifics of your current HPC environment(s) (e.g., number of machines, general architecture)?
  + *Follow-up:* Is there a preference for demonstrating cloud neutrality, or a focus on specific cloud environments currently in use (e.g., AWS presence, potential shift to Azure)?
* **Hardware Acceleration:** What are the requirements for hardware accelerators (e.g., GPUs) for AI/ML training and inference, and how are they currently leveraged?
  + *Follow-up:* Are you interested in demonstrating how OpenShift can provide seamless access to GPUs for containers, similar to how applications are deployed directly on hardware?
* **Orchestration and Control Plane:** How important is it to demonstrate a single "control plane" for managing UFS deployments across multiple HPC clusters and cloud environments?

### **5. User Support, Community, and Collaboration**

* **Workshop Support:** How can OpenShift's capabilities (e.g., "dev spaces," templated environments) specifically address the challenge of setting up instances for a large number of students (e.g., 30-40) for workshops and training sessions?
* **Community Infrastructure:** What are the most critical aspects of the "public-facing community modeling infrastructure" (e.g., EPIC Community Portal (ECP), GitHub discussions, user forums, technical FAQs, tutorials) that the POC should enhance?
* **External Collaboration:** How can the POC facilitate "community contributions" and "co-development" with external partners (universities, private companies, international agencies) in a more streamlined way?

### **6. Security and Compliance**

* **Federal Authorizations:** What specific federal authorizations (e.g., FedRAMP Moderate/High, DOD SRG IL2/4/5) are most critical for the UFS environment, and how can the POC demonstrate compliance?
* **Risk Management Framework:** How can the POC specifically address alignment with the NIST SP800-53-r5 risk management framework for services and products developed within NOAA system boundaries?
* **Supply Chain Security:** Are there specific concerns regarding compliance with prohibitions on certain telecommunications equipment, software (e.g., Kaspersky), or supply chain security orders (FASCSA) that need to be explicitly demonstrated or validated in the POC?

### **7. POC Logistics and Success Measurement**

* **Scope and Incremental Value:** What is the most feasible "very simple application" or "simple DA demo" that can be used as an initial incremental step for the POC to demonstrate immediate value?
  + *Follow-up:* Are there specific scenarios (e.g., data assimilation (DA) demos using Jupyter, comparison of physically-based vs. data-driven models, or integration of JEDI) that would be highly impactful to demonstrate?
* **POC Success Criteria:** How will the success of this POC be concretely measured and evaluated by NOAA EPIC? Is it primarily a technical validation, a demonstration for leadership, or a stepping stone for wider adoption?
* **Timeline and Resources:** What is the desired timeframe for this POC to take place, especially considering upcoming events like UIFCW25 or contract transitions (EPIC v1 to EPIC v2)?
  + *Follow-up:* What internal resources (e.g., code managers, SMEs, access to existing environments, Docker images) can be allocated to support the Red Hat team for the POC?
* **Collaboration Model:** Given current contractual constraints, what is the preferred and feasible model for continued collaboration during the POC (e.g., regular weekly meetings, joint troubleshooting sessions, or a more hands-on workshop either at NOAA or Red Hat facilities)?

POC Design

<https://github.com/NOAA-EPIC/CADRE-DA-training>

Docker image is located in this git repository

Global web cast

Low resolution modeling

Emphasize this is an enterprise framework while showing functionality

15 min talk to cover AI

Goal??

Pull image

Create singularity image

Run the image in OCP

Issue:

6 months to deploy a stack

Why are your docker images 7.6G