



Deploying Cloud-Based Automated Continuous Integration to Assist in the Community Development of GFDL's Finite Volume Cubed-Sphere Dynamical Core

Lauren Chilutti^{1,2}, Thomas Robinson¹, Rusty Benson¹, Lucas Harris¹, Luis Sal-Bey *in memoriam*^{1,2}

¹NOAA/Geophysical Fluid Dynamics Laboratory, Princeton, NJ, ²SAIC

Introduction

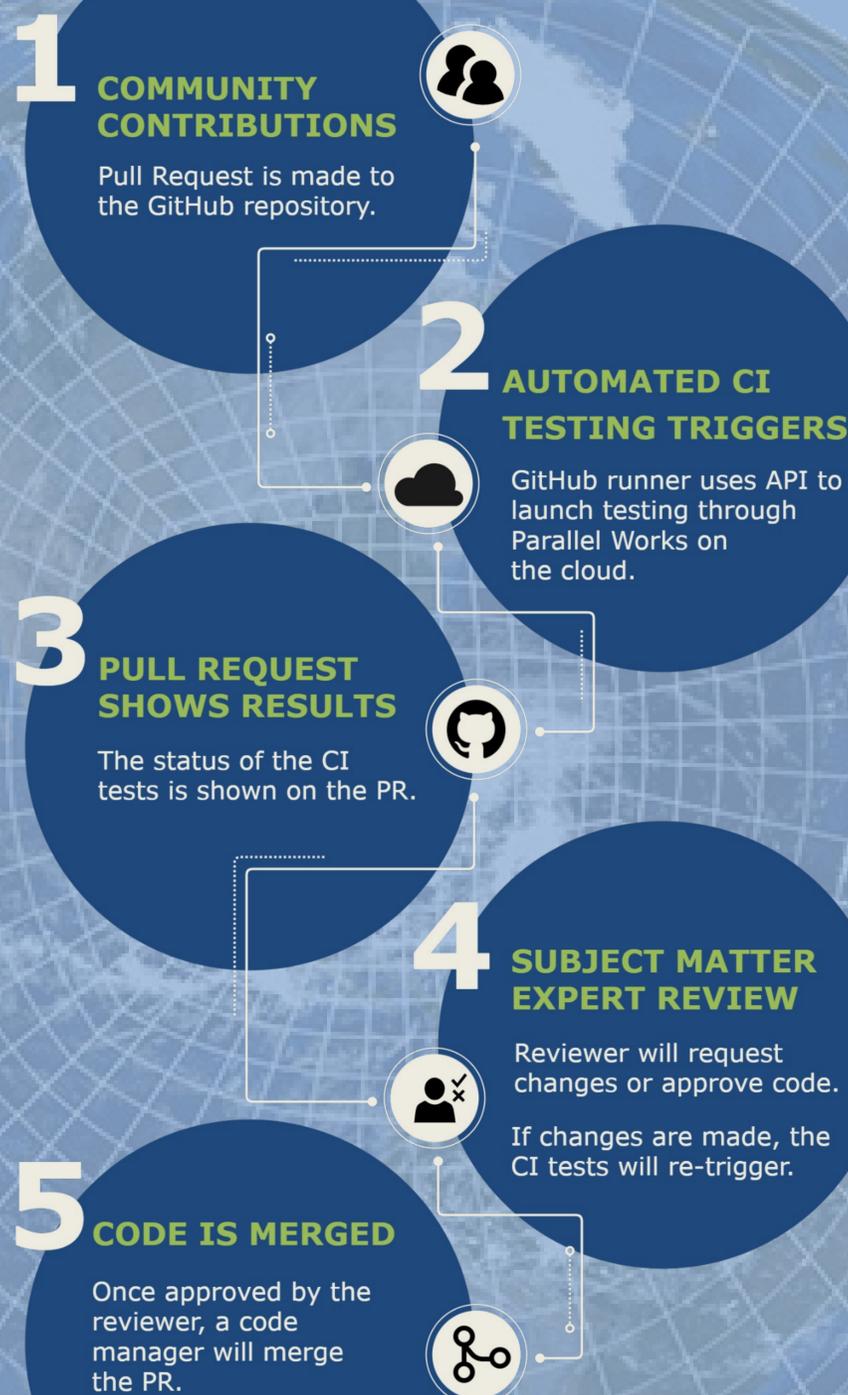
Automated testing of contributions made to the GFDL Finite-Volume Cubed Sphere Dynamical Core (FV3) public repository is paramount for ensuring the integrity of the many earth-system models and forecasting applications using FV3 as a dynamical core. We have designed a continuous integration (CI) approach for the FV3 dynamical core GitHub repository.

Our Process

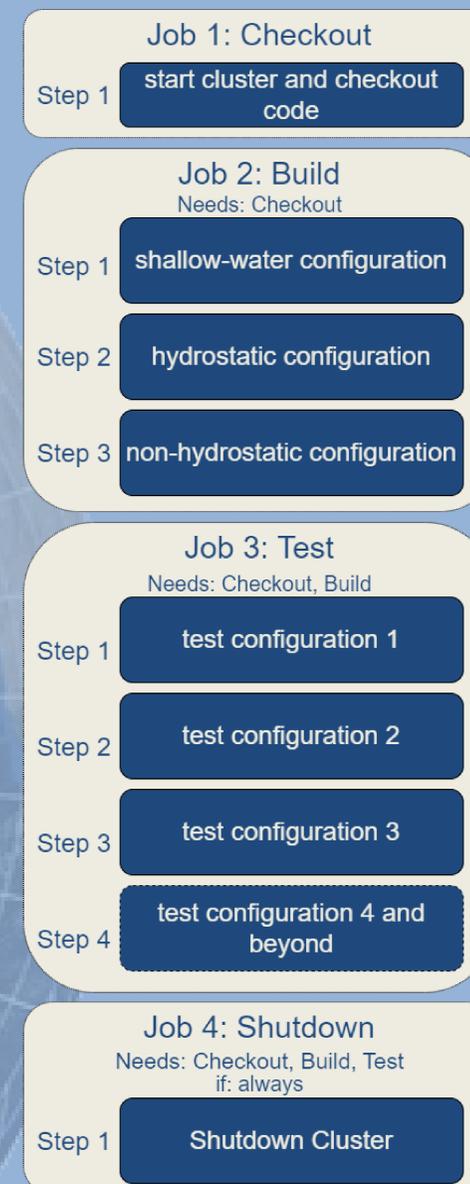
- Configure the NOAA Parallel Works cloud gateway for communication with GitHub as a self-hosted runner
- Configure a NOAA Parallel Works cloud cluster exclusively for the FV3 CI. We use a Microsoft Azure HPC cluster running an AMD64 architecture containing 96 compute nodes.
- Install a NOAA-GFDL Singularity container on the FV3 CI cluster ensuring that this is stored in a permanent file storage directory.
- Revise existing build and test scripts to work inside of the cloud/container environment.
- Develop an API to communicate with the NOAA Parallel Works cloud gateway. The API will start and shut down the FV3 CI cluster as well as execute checkout, build, and test scripts on the FV3 CI cluster.
- Generate a GitHub workflow YAML file. We specify that the workflow runs on our self-hosted runner for all Pull Requests targeting the main branch and executes the FV3 CI API.



to see our GitHub repository



Actions pipeline defined by the workflow YAML file



Conclusion

Implementing a continuous integration workflow in the GFDL FV3 Dynamical Core required adapting the existing build and test system for more general use, configuring a cloud computing platform, and using GFDL containers. This has enhanced the code review process and ensures the integrity of the code.