This work is supported by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics, under contract DE-AC02-06CH11357, and Argonne's LDRD program.

SOFTWARE R&D FUNDED BY ARGONNE LDRD (ULAB)

Argonne NATIONAL LABORATORY 75 1946–2021

AUTOMATED WORKFLOWS FOR DETECTOR DESIGN

SYLVESTER JOOSTEN
sjoosten@anl.gov
On behalf of the Argonne EIC LDRD team

WHITNEY ARMSTRONG
warmstrong@anl.gov

Software & Computing Roundtable
Tuesday, May 4, 2021

A NEW DETECTOR FROM SCRATCH?

From the EIC Yellow Report to an optimized EIC detector

η	Nomenclature	Tracking						Electrons and Photons			π/K/p		HCAL		Muons
		Resolution	Relative Momentun	Allowed X/X_0	Minimum-p_T (MeV/c)	Transverse Pointing Res.	Longitudinal Pointing Res.	Resolution σ_E/E	PID	Min E Photon	p-Range	Separation	Resolution σ_E/E	Energy	
< -4.6	Low-Q2 tagger														
-4.6 to -4.0		Not Accessible													
-4.0 to -3.5			Reduced Performance												
-3.5 to -3.0	Backward Detector		σ_p/p ~ 0.1%×p⊕2%	~5% or less	150-300			1%/E ⊕2.5%/√E ⊕1%	π suppression up to 1:10^4	20 MeV	≤ 10 GeV/c	≥ 3σ	50%/√E ⊕ 10%	~500MeV	Muons useful for background suppression and improved resolution
-3.0 to -2.5															
-2.5 to -2.0			σ_p/p ~ 0.02% × p ⊕ 1%												
-2.0 to -1.5						dca(xy) ~ 40/p_T μm ⊕ 10 μm	dca(z) ~ 100/p_T μm ⊕ 20 μm	2%/E ⊕(4-8)%/√E ⊕2%	π suppression up to 1:(10^3-10^2)	50 MeV					
-1.5 to -1.0															
-1.0 to -0.5	Barrel		σ_p/p ~ 0.02% × p ⊕ 5%		400	dca(xy) ~ 30/p_T μm ⊕ 5 μm	dca(z) ~ 30/p_T μm ⊕ 5 μm	2%/E ⊕(12-14)%/√E ⊕(2-3)%	π suppression up to 1:10^2	100 MeV	≤ 6 GeV/c		100%/√E ⊕ 10%		
-0.5 to 0.0															
0.0 to 0.5															
0.5 to 1.0															
1.0 to 1.5	Forward Detectors		σ_p/p ~ 0.02% × p ⊕ 1%		150-300	dca(xy) ~ 40/p_T μm ⊕ 10 μm	dca(z) ~ 100/p_T μm ⊕ 20 μm	2%/E ⊕(4^-12)%/√E ⊕2%	3σ e/π up to 15 GeV/c	50 MeV	≤ 50 GeV/c		50%/√E ⊕ 10%		
1.5 to 2.0															
2.0 to 2.5															
2.5 to 3.0															
3.0 to 3.5															
3.5 to 4.0	Instrumentation to separate charged particles from photons		Reduced Performance												
4.0 to 4.5		Not Accessible													
> 4.6	Proton Spectrometer														
	Zero Degree Neutral Detection														

+ new developments

Detector & reconstruction requirements

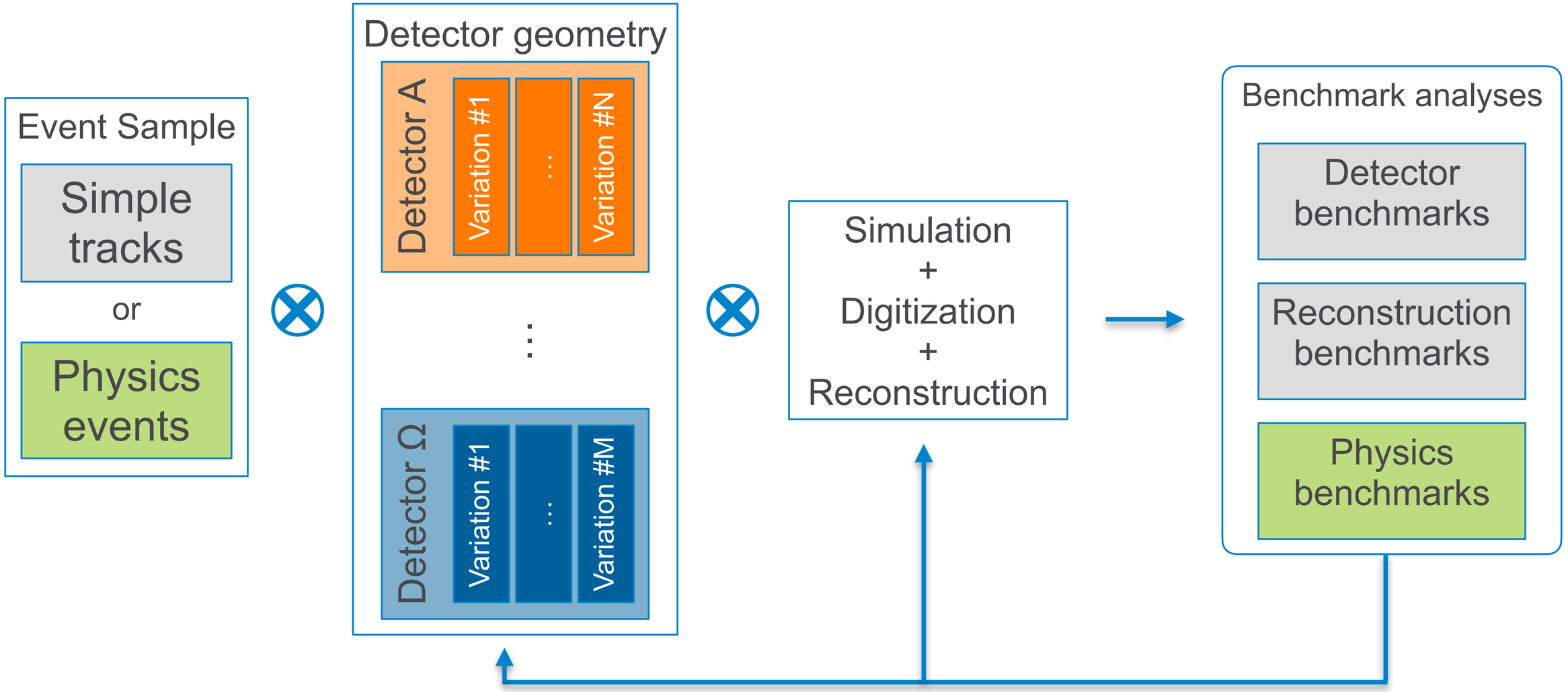
Extensive list of key performance parameters inform detector choices. This table of requirements could be interpreted as a series of automized tests that a detector implementation needs to pass.

Physics requirements

Detector design has to enable many key physics measurements, while being flexible enough to accommodate new developments through the next 2 decades

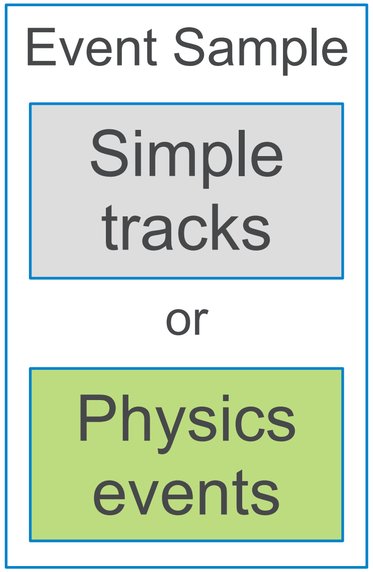
Enabled by modular, interoperable toolkit

DETECTOR OPTIMIZATION AS AN AUTOMIZED WORKFLOW



EVENT GENERATION

Standardized around HepMC3 data format



- Validate detector and benchmark reconstruction chain against YR requirements (using particle guns)

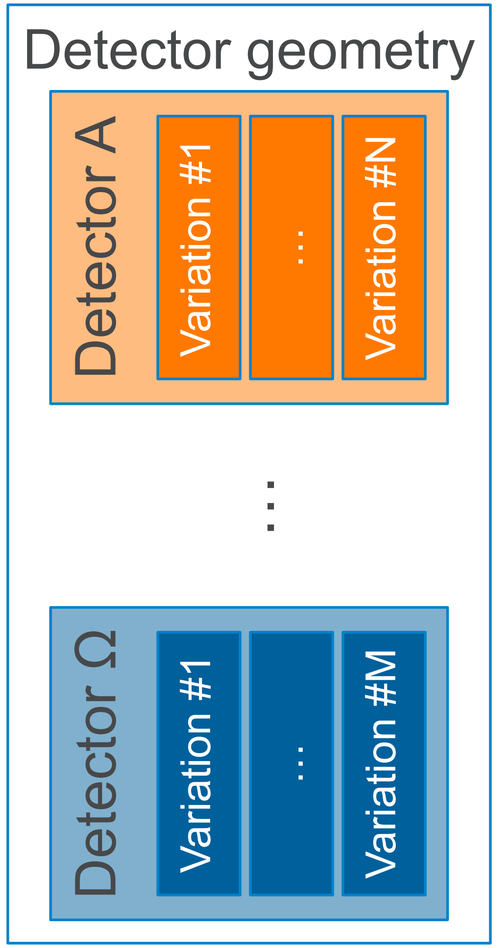
- Characterize and benchmark detector setup for desired physics observables (using physics event generators)

- Towards a fully integrated setup for on-demand event generation to cover all NAS/YR requirements for EIC
- Software chain standardized around the HepMC3 format

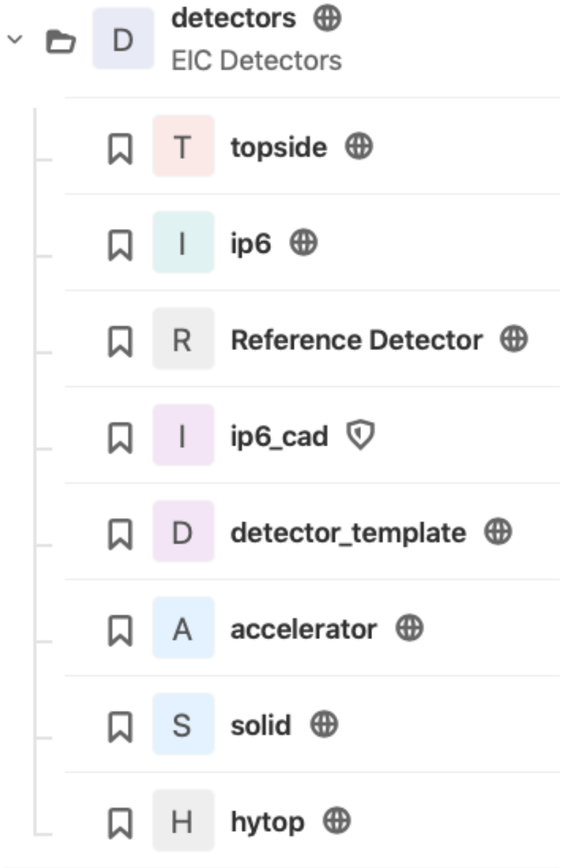
https://github.com/AIDASoft/DD4hep
https://eicweb.phy.anl.gov/EIC/NPDet
https://eicweb.phy.anl.gov/EIC/detectors

DETECTOR GEOMETRY

Parametrized detector implementations with DD4hep + NPDet



- Parametrized detector description defined using DD4hep
- NPDet: an extension for DD4hep that adds parametrized detectors for NP experiments (e.g. Cherenkov counters) and various DD4hep based tools (visualization, conversion, inspection, …)
- **Single, unique source for geometry**: dispatches full GEANT4 simulation and provides geometry for reconstruction
- Aim towards library of configurable detector options to feed into benchmark & optimization process.



MAIN TOOLKIT

Simulation/digitization/reconstruction

Simulation
+
Digitization
+
Reconstruction

DD4hep: Geant4 geometry fully defined through detector plugin library, provides wrappers to run Geant4

ACTS: Experiment-independent tracking toolkit developed for HEP in modern C++, ACTS' geometry constructed from DD4hep via plugin

GAUDI: Generic open project for building event processing frameworks. Enables modern task-based concurrent execution in a heterogeneous computing environment.

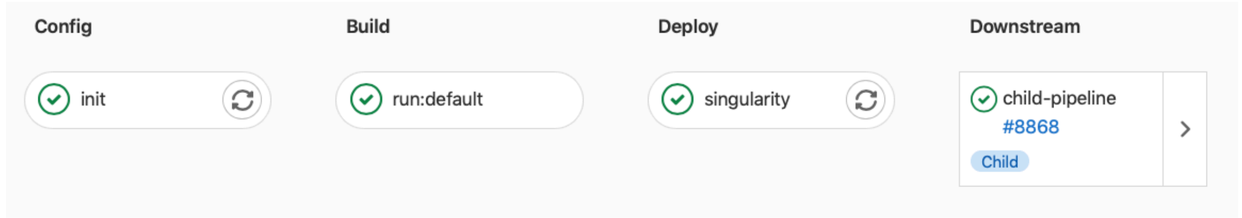
Podio: Robust data model definition to cross the boundaries between the tools

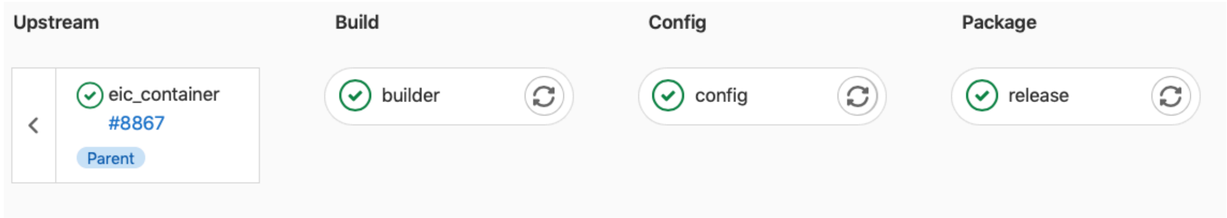
Project Juggler: Prototype event processing framework for EIC, tying together GAUDIO, ACTS, DD4hep and Podio. Used for the digitization and reconstruction step.

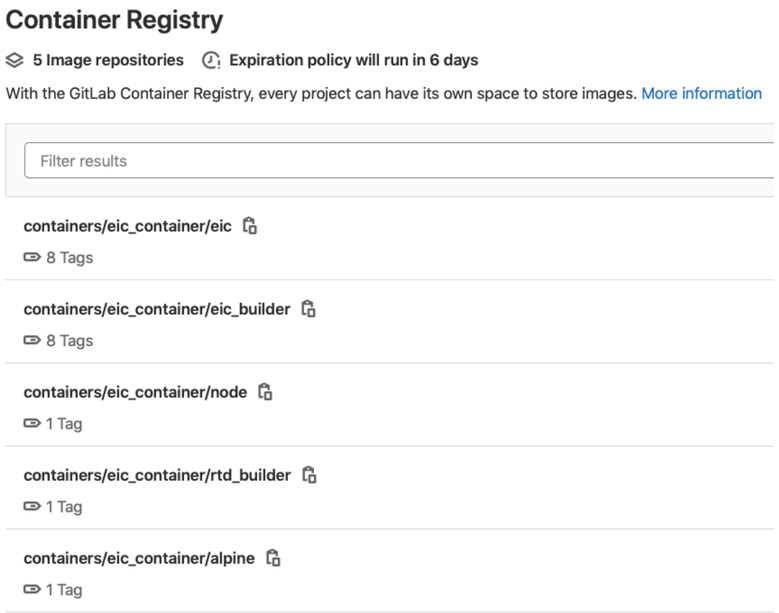
Note similarity with key4hep approach!

BUILDING THE SOFTWARE

Containerization and deployment







GitLab

Leverage own GitLab server instance (eicweb), with continuous integration and dedicated build cluster, to automatically build container

Spack

Reproducible container build with Spack

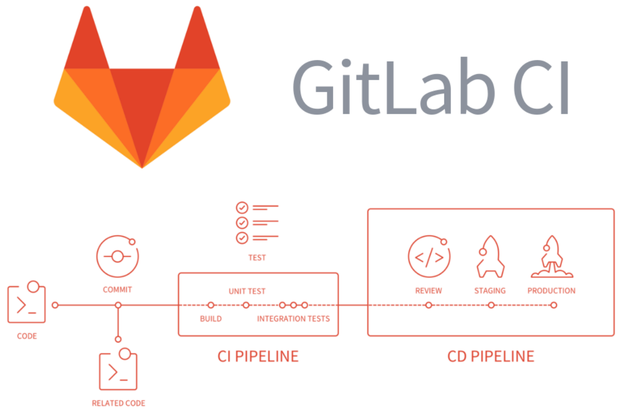
docker

Docker container used in CI toolchain (and local development on MacOS)

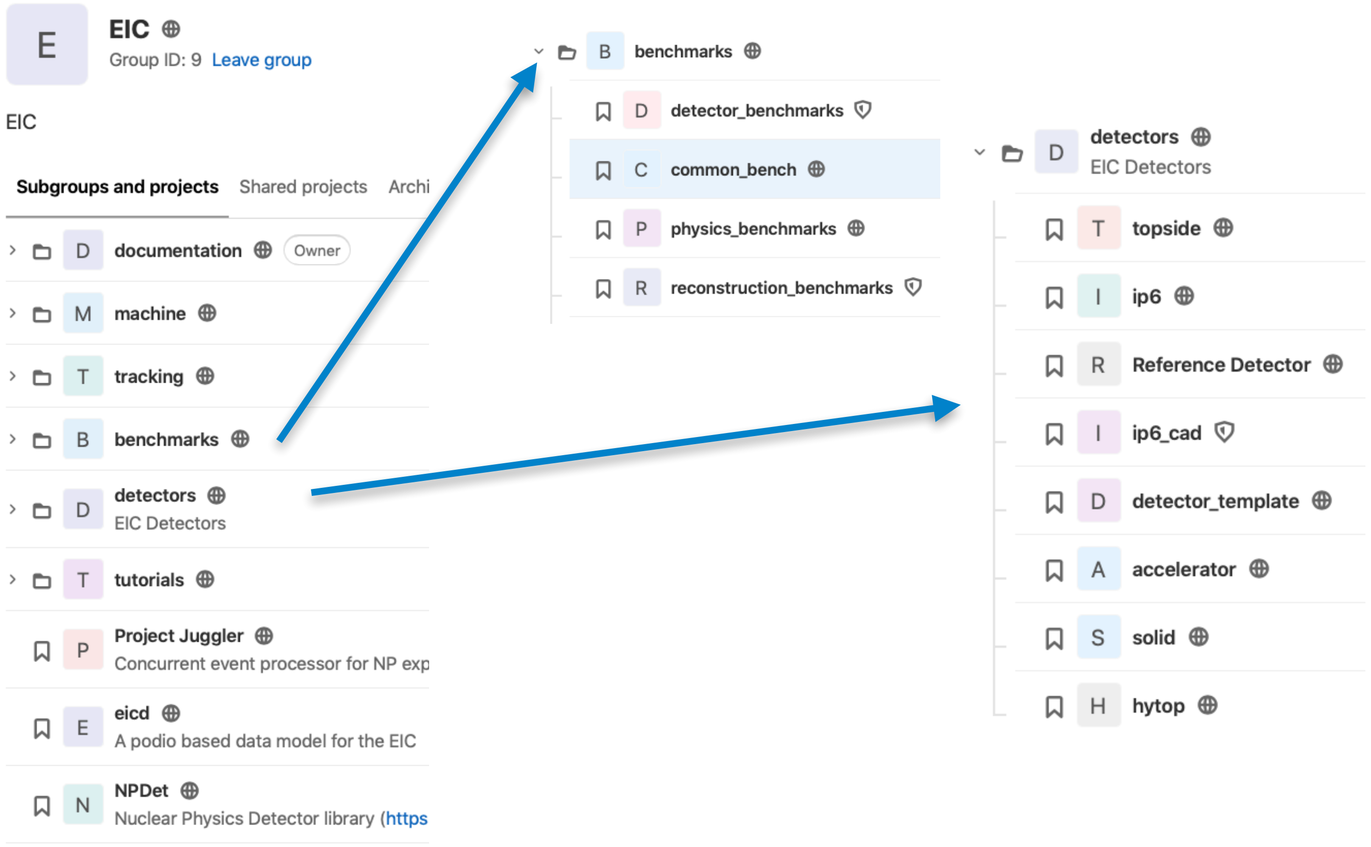
Singularity container used for deployment to HPC environments (and local development)

NEW PARADIGM IN SIMULATION WORKFLOW

From classic CI/CD to detector and physics benchmarks



- **CI/CD**: continuous integration & Deployment
- **Pipeline**: Collection of interdependent jobs that run in a certain sequence
- **Job**: A task (script) that runs in a CI pipeline
- **Artifact**: File(s) produced by a job uploaded to GitLab
- **Pipeline trigger**: Connecting pipelines

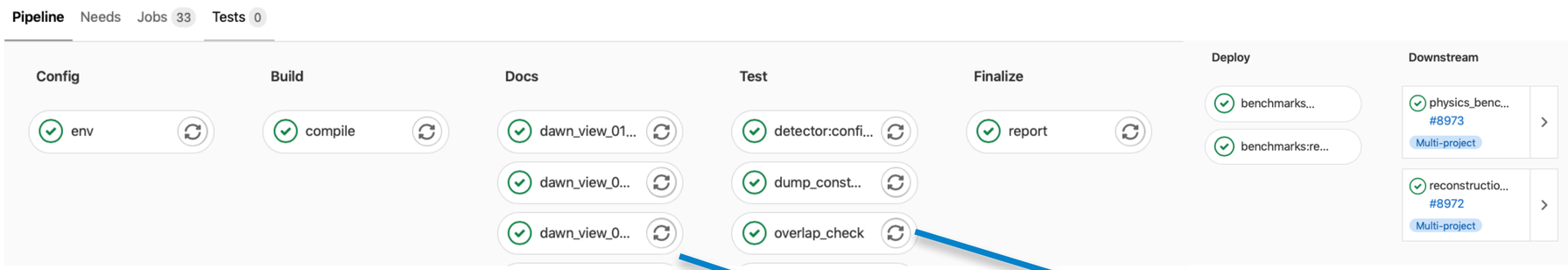


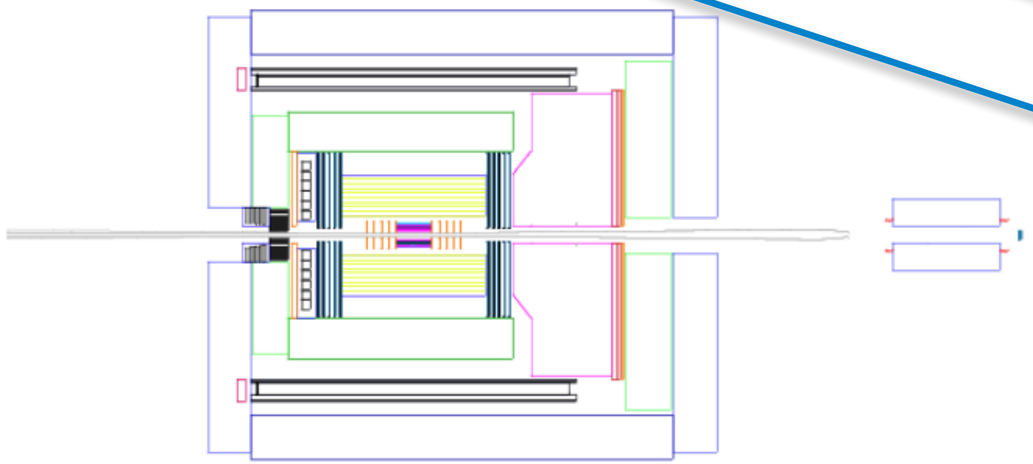
Many repos chained together, organized with GitLab groups

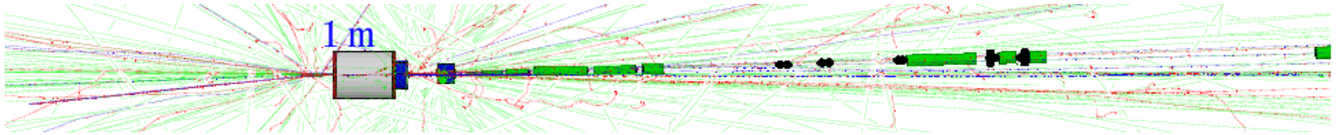
https://twiki.cern.ch/twiki/bin/view/CLIC/DawnVisualization

PIPELINE EXAMPLES

Automatic detector tests and documentation

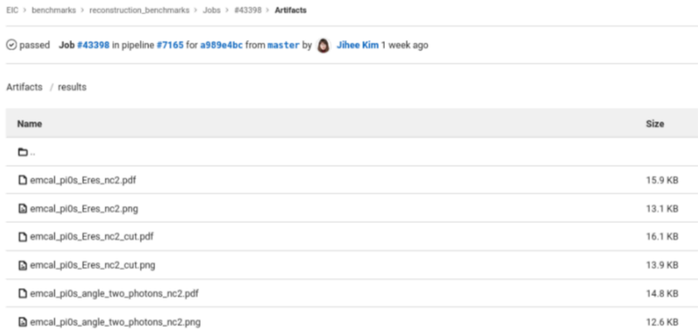






Geometry overlap checks running as part of every merge request

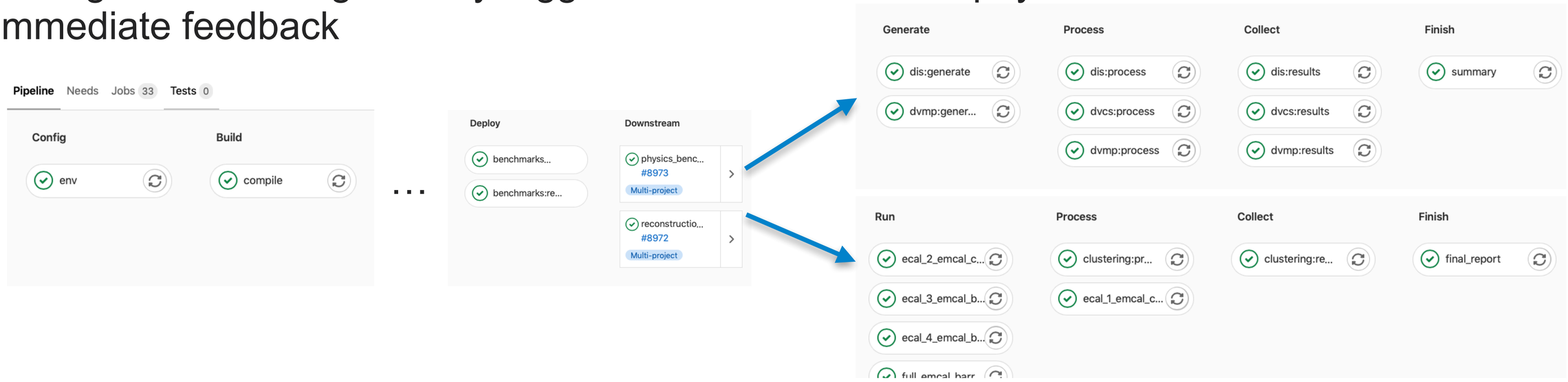
Automatic visualizations for detector geometries, saved as job artifacts (browsable!)



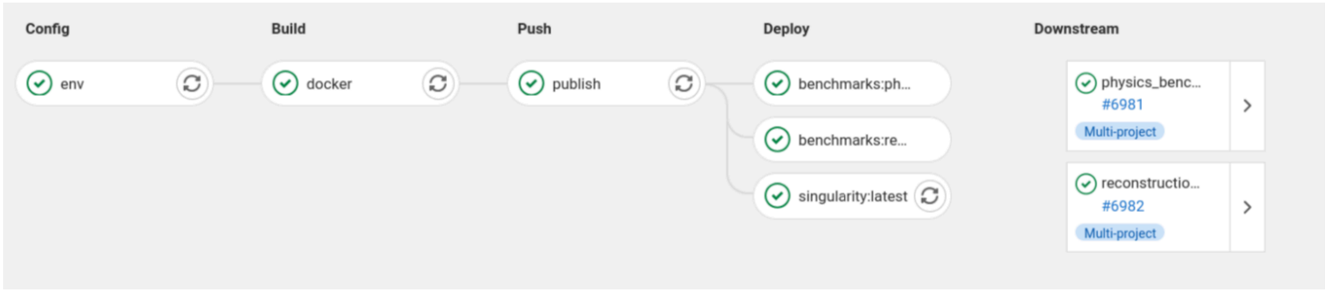
PIPELINE TRIGGERS

Connecting pipelines for immediate feedback

Change in detector geometry triggers reconstruction and physics benchmarks for immediate feedback



Also dispatch from Jugger CI: Due to merge request trigger, we can ensure proper operation of new/updated reconstruction algorithms



OPEN CHALLENGES

- Need full library of Monte Carlo generators that cover entire YR program
- Automatic publication/documentation of performance metrics to webpage for discoverability
- Compiling/combining metrics from benchmarks to guide optimization
- Distributed data persistency required, as GitLab artifact approach does not scale for large simulations
- Dispatch to HPC integrated in workflow (custom singularity gitlab runner? K8s? Manual pre-computations?)

EXAMPLE:
SIMPLE BROWSER-BASED WORKFLOW

BROWSER-BASED WORKFLOW EXAMPLE

Let’s change VertexTrackerRadius1 from 30mm to 29mm

vertex_tracker.xml 8.8 KB

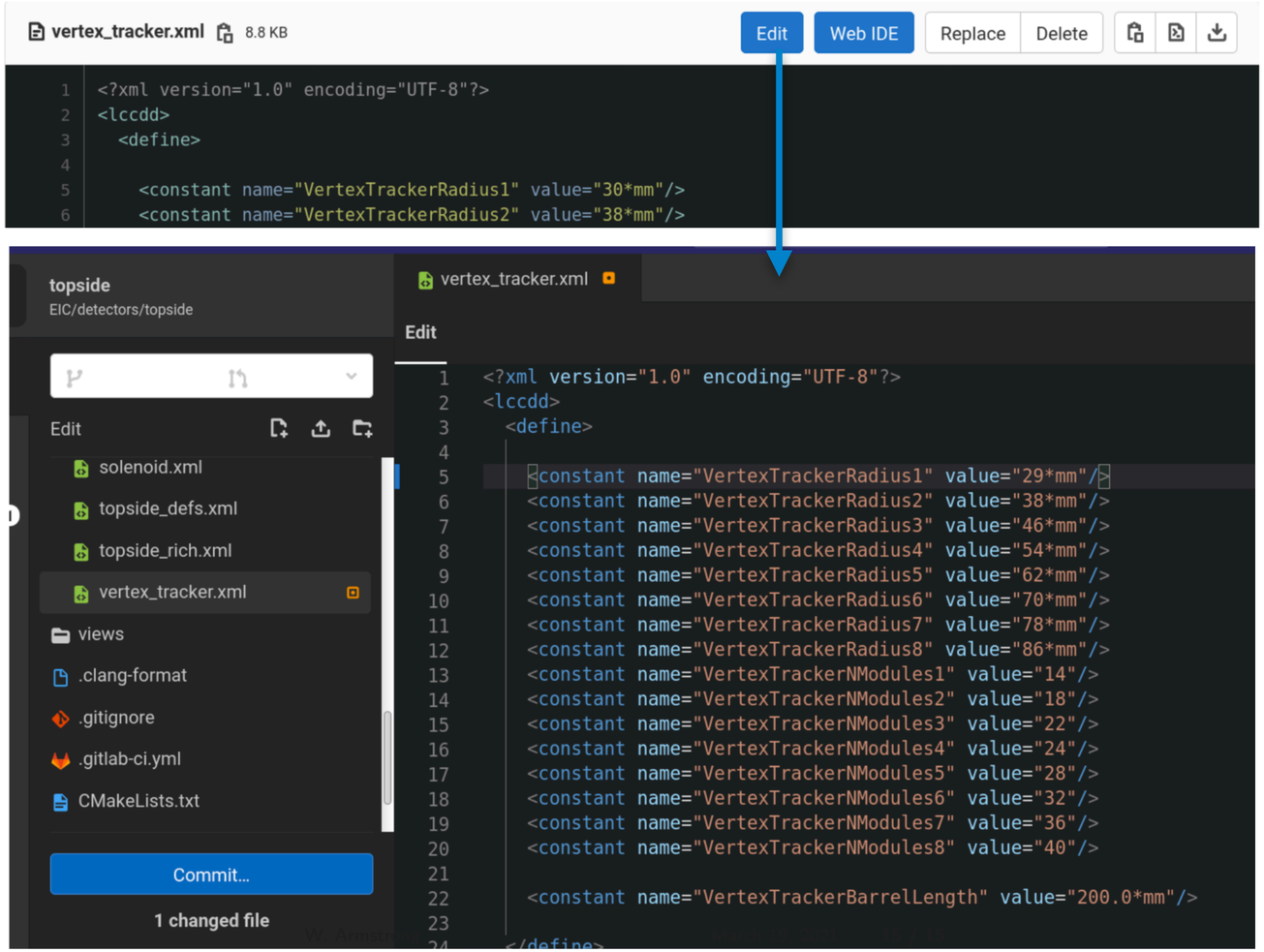
Edit | Web IDE | Replace | Delete

```
<?xml version="1.0" encoding="UTF-8"?>
<lccdd>
  <define>

    <constant name="VertexTrackerRadius1" value="30*mm"/>
    <constant name="VertexTrackerRadius2" value="38*mm"/>
    <constant name="VertexTrackerRadius3" value="46*mm"/>
    <constant name="VertexTrackerRadius4" value="54*mm"/>
    <constant name="VertexTrackerRadius5" value="62*mm"/>
    <constant name="VertexTrackerRadius6" value="70*mm"/>
    <constant name="VertexTrackerRadius7" value="78*mm"/>
    <constant name="VertexTrackerRadius8" value="86*mm"/>
    <constant name="VertexTrackerNModules1" value="14"/>
    <constant name="VertexTrackerNModules2" value="18"/>
    <constant name="VertexTrackerNModules3" value="22"/>
    <constant name="VertexTrackerNModules4" value="24"/>
    <constant name="VertexTrackerNModules5" value="28"/>
    <constant name="VertexTrackerNModules6" value="32"/>
```

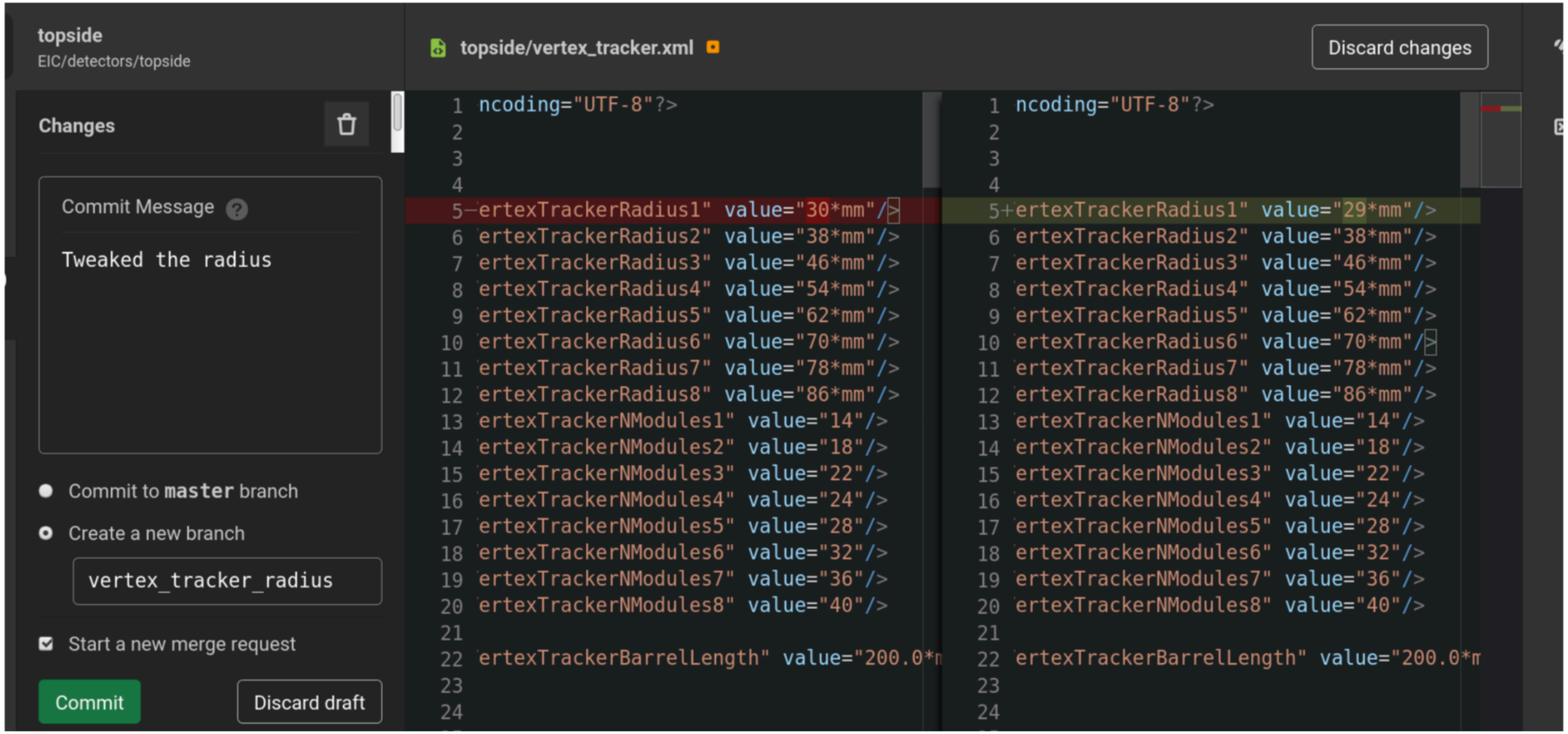
BROWSER-BASED WORKFLOW EXAMPLE

Simplest workflow uses the browser-based GitLab IDE



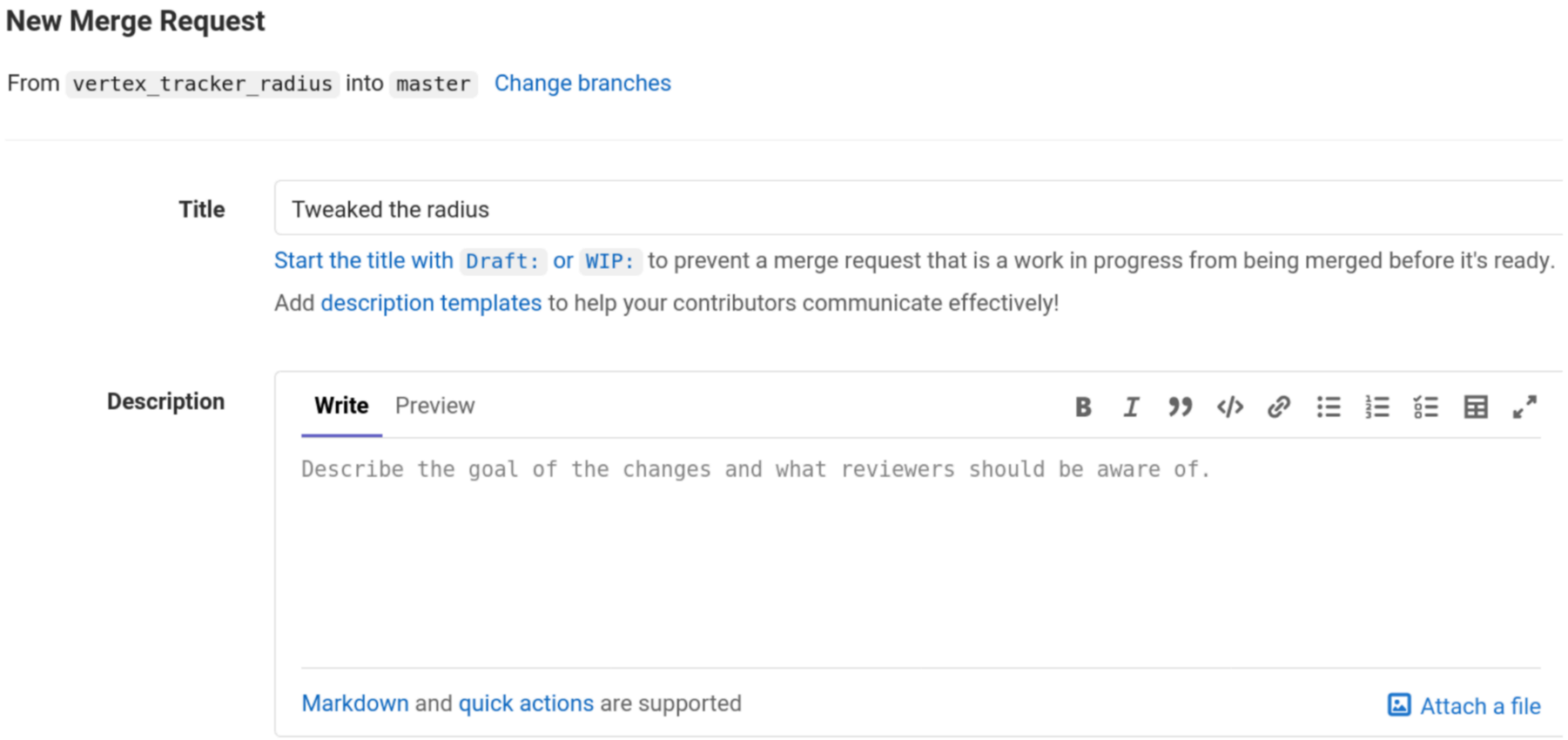
BROWSER-BASED WORKFLOW EXAMPLE

Commit changes to a new branch, and…



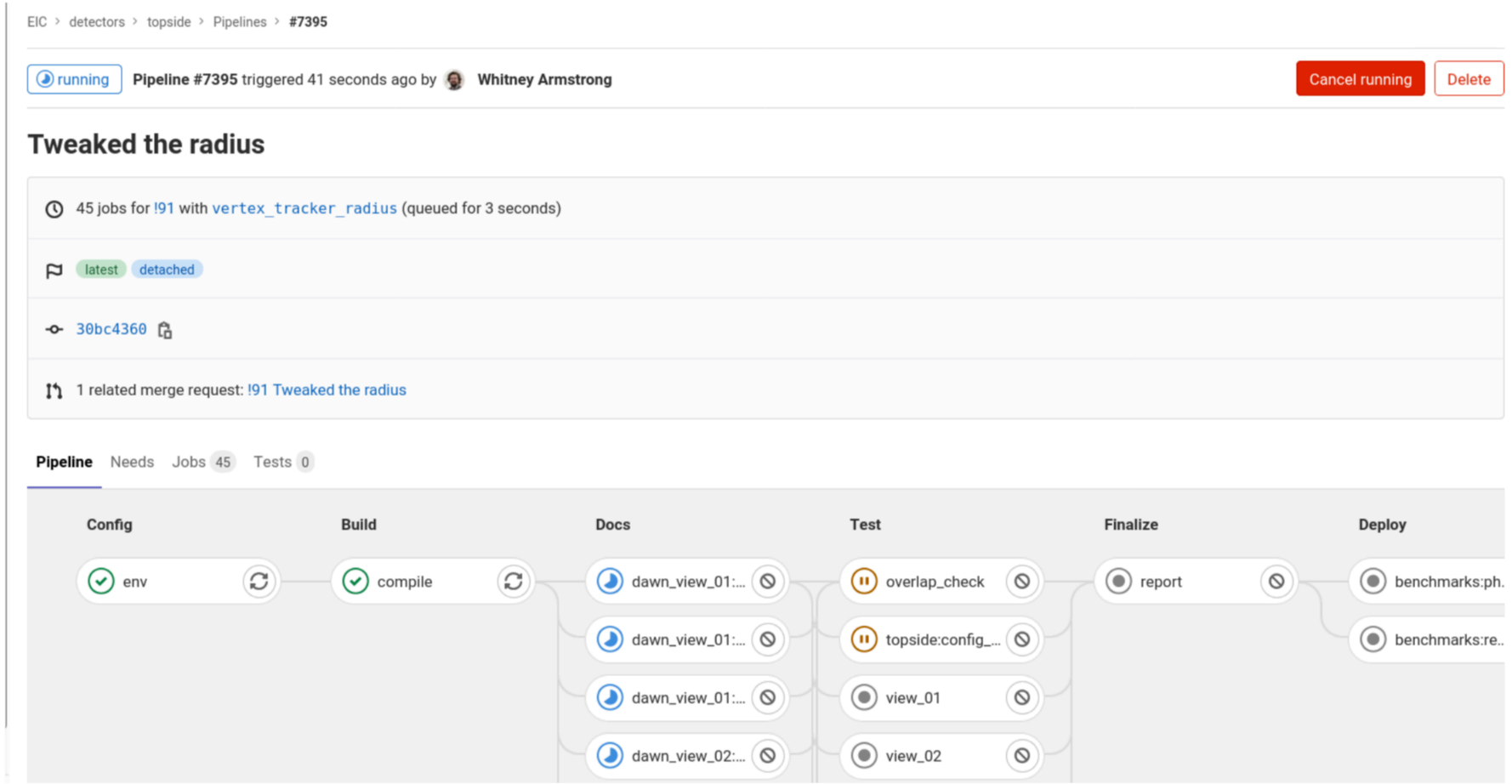
BROWSER-BASED WORKFLOW EXAMPLE

…create a new Merge Request



BROWSER-BASED WORKFLOW EXAMPLE

…the Merge Request then triggers the CI.



ALL WITHOUT LEAVING THE BROWSER OR INSTALLING ANY SOFTWARE LOCALLY!

Our software R&D in the context of the EICUG software EoI

A WORKFLOW-CENTRIC SOFTWARE APPROACH FOR EIC

- **Workflows -** *leveraging the power of GitLab CI*
  - ✓ Automized simulation-reconstruction-analysis pipelines
  - ✓ Low bar of entry, were able to onboard external collaborators (students!) with minimal training, already doing valuable work.
  - ✓ **next:** build out analysis portfolio, better dispatch to HPC, persistent distributed data storage, “publish” benchmark results through GitLab Pages

- **Detector simulations**
  - ✓ Geometry and simulation tooling around DD4hep
  - ✓ **next:** develop library of parametrized detector concepts, accelerate simulation with GANs
- **Reconstruction**
  - ✓ Functioning generic tracking with ACTS (and Gaudi)
  - ✓ **next:** reconstruction for other subsystems, improve task-based concurrency, accelerate part of workflow on GPUs, or with AI.
- **Data model**
  - ✓ Data model definition with podio
  - ✓ **next:** optimize and freeze exact data model, together with community
- **Explore user-centered design**
  - ✓ Modular workflow with low bar of entry for analysis
  - ✓ **next:** automatic publication of pipeline results to website
- **Data analysis preservation:**
  - ✓ Analysis as part of benchmark portfolio pipeline
  - ✓ **next:** website to aggregate analysis results and monitor quality of results (in the vein of benchmarks/tests).
- **Discoverable software**
  - ✓ Software pipeline fully containerized, software stack built and managed with Spack
  - ✓ **next:** integration of newer software tools with Spack

THE END