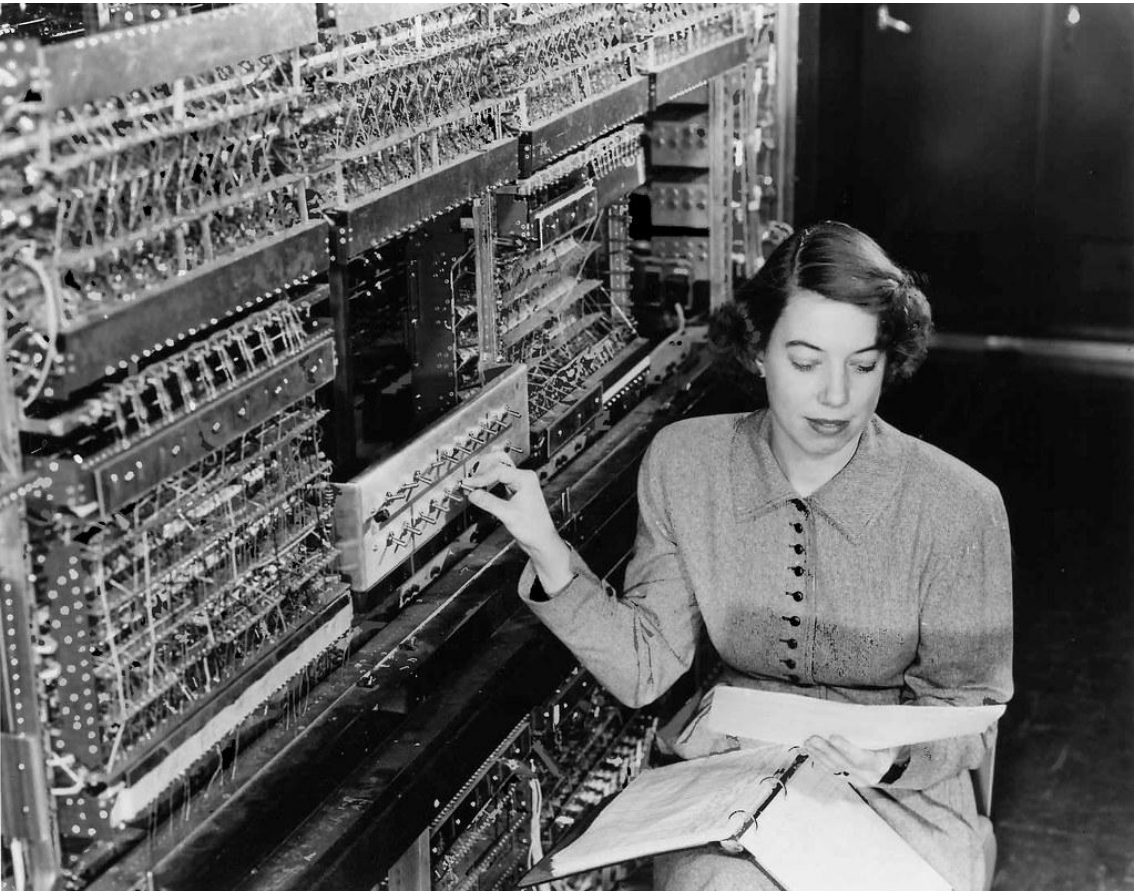
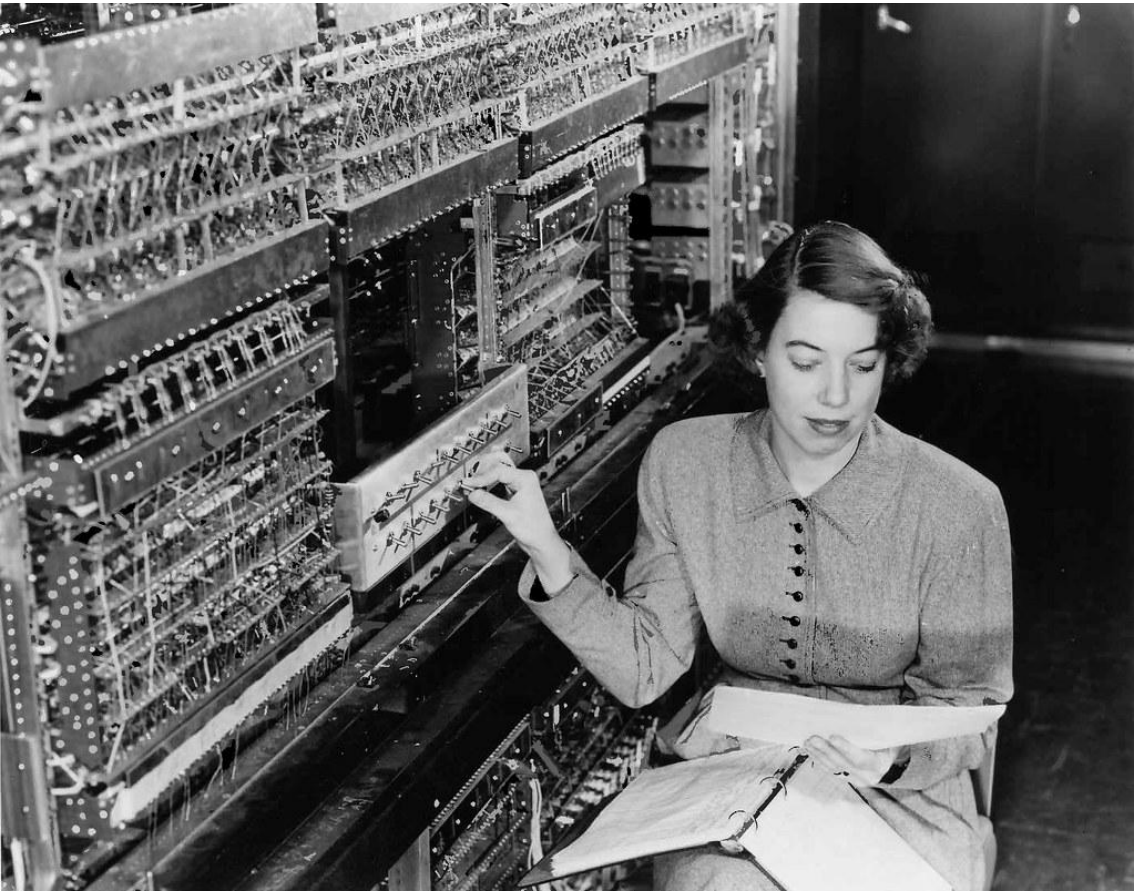


Gitlab and the EIC project detector:

“Code Repository” Discussion

Wednesday, May 25th 2022

Whitney Armstrong (Argonne National Laboratory) ,
Sylvester Joosten (Argonne National Laboratory),
Wouter Deconinck (The University of Manitoba

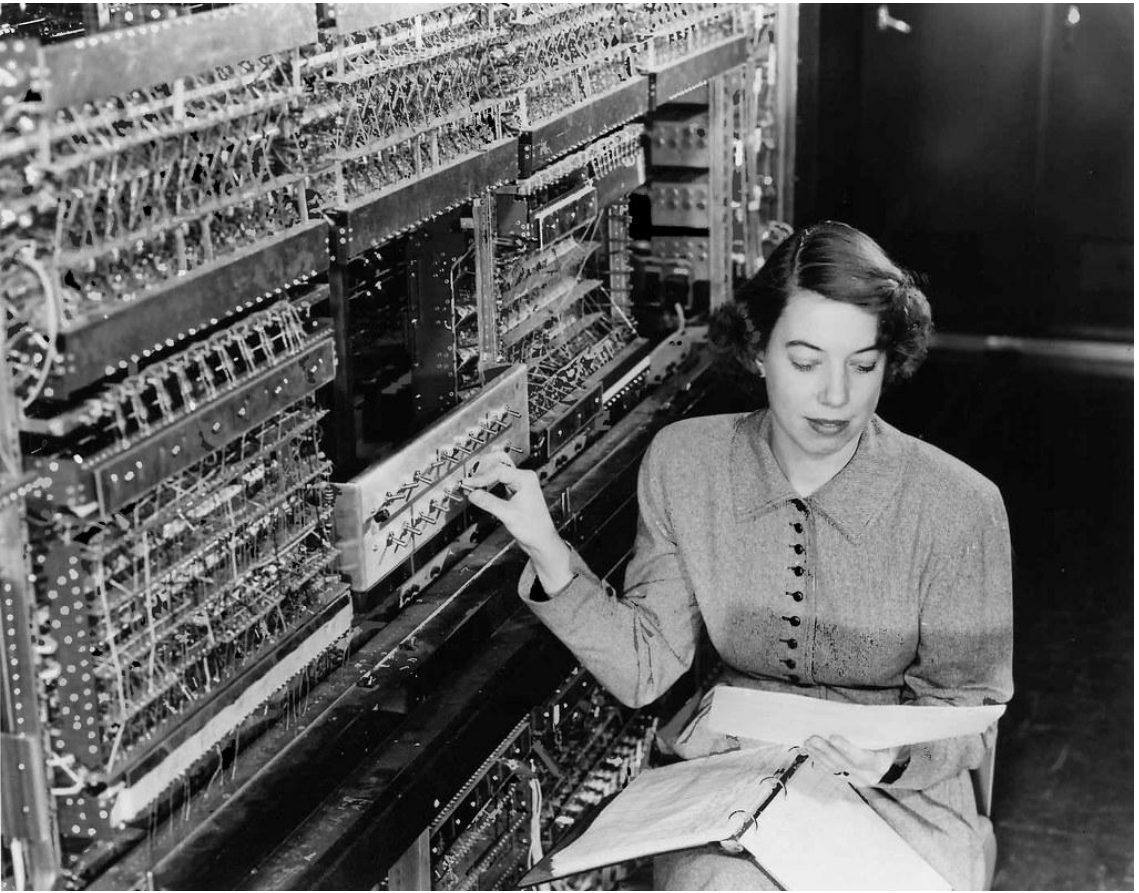


~~Gitlab and the EIC project detector:~~

~~"Code Repository" Discussion~~

Wednesday, May 25th 2022

GitLab



Productive Gitlab workflows for the EIC Project Detector

Wednesday, May 25th 2022

GitLab

Introduction

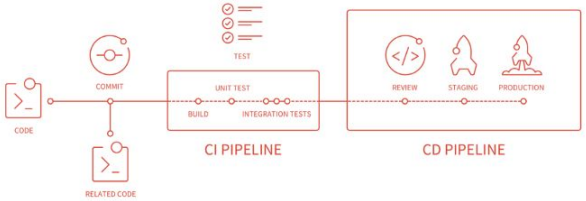
Tasked with discussing gitlab as a “Code Repository” decision

- *We really want to decide is:* **what collaboration and development platform should we use**?
- That is, **how will we work together** on software for the EIC?
- Ultimately we are deciding on **workflows**
- This means considering CI/CD and containerization too (I have tried to keep this to a minimum)
- I recommend we use eicweb.phy.anl.gov, anybody can signup for an account using their institutional email (ie not gmail).

Gitlab workflows for the EIC Project Detector

- Gitlab has too many features to discuss them all. It has issues, milestones, groups and subgroups (own milestones), issue board tracker, wiki, permissions configuration for nearly everything including pipeline triggers…
- Permissions and roles
  - ‘Admins’ maintain the server, these would be a few experts.
  - ‘Owners’ manage groups and projects, traditional thought of as the role of admin
  - ‘Maintainers’ have slightly less permissions than owners
  - ‘Developer’, ‘Reporter’, and ‘Guest’ - See documentation for details
- Gitlab hits all the requirements I have seen.
  - ✅ Cloud service accessible from anywhere in the world
  - ✅ Does not require a paid account for each user
  - ✅ Top-level repositories can be configured with different access policies ranging from world-readable to private (with access only for select users)
  - ✅ Supports Continuous Integration (CI)
  - ✅ (self hosted) Non-restrictive limits:
    - >=1000 repositories
    - >=1TB (w/ ability to increase as needed w/o significant additional cost)
    - >=10TB/mo (w/ ability to increase as needed w/o significant additional cost)
- I would also add the requirements
  - ✅ Low entry barrier – it should be easy to use and quickly make contributions
  - ✅ Integrated container registry and customizable CI executors (eg gitlab-runners)

GitLab CI

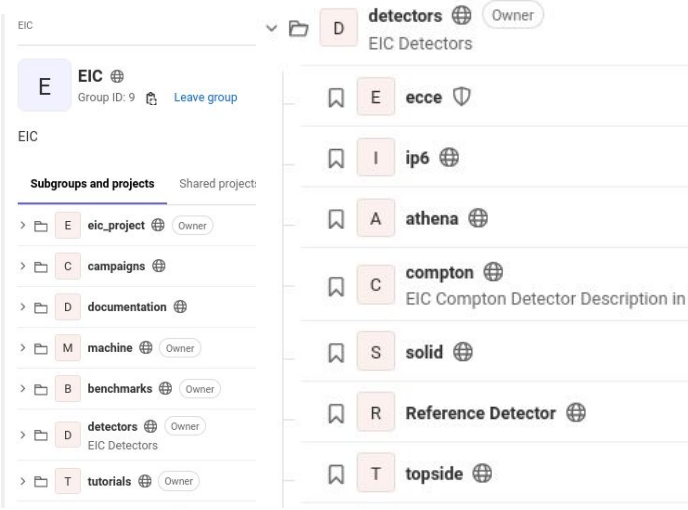


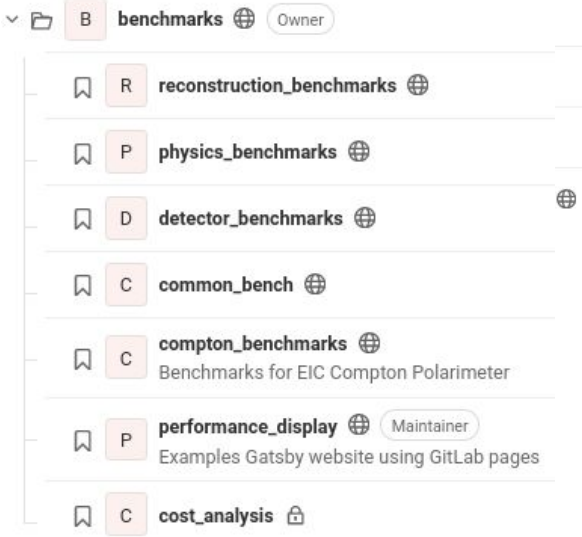
Software and Simulation R&D with eicweb.phy.anl.gov

- What is eicweb?
  - Self-hosted gitlab server at Argonne
  - Dedicated to EIC user community for software
  - More than just a software repo: **eicweb is a software and simulation R&D platform**
- How did we use eicweb?
  - Our use and strategy for eicweb evolved
  - Initially just a gitlab repo with issues, versions, etc…
  - Started using CI in typical ways: compile checks, container builds
  - Our use case for the EIC is different in many ways so we started doing new things with CI/CD pipelines and artifacts
  - The monolithic “Code Repository” model was dropped for a growing collection of groups and sub-groups of smaller repos.
  - Containerization matured in parallel (docker and singularity builds)
- We learned a lot while developing the ATHENA proposal and have identified many areas (big and small) for improvement
  - Eicweb could be characterized as an ‘analysis facility’ and is the direction we are headed

Eicweb Repositories and pipelines

- Initially nearly all detector/geant4 code was in **`NPDet`**
  Quickly realized NPDet was really temporary:
  - Addons could (should) be pushed to upstream projects (eg dd4hep or podio)
  - Components should be separated (eg ip6 geometry and central detector)
- NPDet is deprecated but highlights how a more granular approach to structuring groups and repositories lends itself to greater clarity.
- The benchmarks group was initially split into detector, reconstruction, and physics repositories
  - A single detector and reconstruction benchmarks repo seemed to work fine.
  - We struggled to fully develop physics benchmarks in a robust way. Likely due to the monolithic approach.
  - Physics benchmarks is better off as a collection of subgroups looking at analysis of multiple but related observables. For example, `EIC/benchmarks/deep_exclusive_benchmarks` might be looking at the combined DVMP and DVCS data with proton and deuteron beams to provide some flavor sep of GPDS/CFFs.
  - Also, getting physics WGs to contribute to the development of benchmarks was difficult.
  - Also, we never succeeded in applying a benchmarking helper library for robust metrics on performance for detector comparisons





Why not other development platforms?

- ❌ Github is a closed source service
- ❌ Subject to github platform changes (microsoft)
- ✅ Self-hosted gitlab releases are frequent and changes well documented
- ✅ Gitlab is constantly improving and new features added
- Gitea would be a good self hosted alternative to github
- Their website provides feature comparisons among the different platforms
- ✅ In the future, eicweb hosting can be relocated or even distributed
- ✅ Gitlab's powerful v4 REST API and is being upgraded to a much more powerful and modern GraphQL API – lots of opportunity for new ideas and development tuned to our unique use case
- ✅ Gitlab is a dedicated cluster of heterogeneous hardware and services backing up the repository which we can grow as needed

What’s coming to eicweb in the future

- We are pushing the limits of the current server configuration
- But we have a much more powerful machine with more storage and faster network connection in a new location. Will migrate servers in the next few weeks (should not be noticed by users).
- Heterogeneous computing and HPC runners.
- **Kubernetes cluster** for software and simulation R&D, extra gitlab features enabled with kubernetes clusters.
- Lots of other ideas not directly related to gitlab but that assume its functionality is available (eg S3 storage for intermediate simulation data cache, maybe rucio)

All you need is a web brows

- Nearly **all software and simulation development** can be **completed** with gitlab **in your browser** (this should be the preferred method over running locally)
- To make a change you only need to know how to use the built-in editor to make a commit (like changing the tracker radius by 1cm)
- Creating a new MR, triggers the pipelines
- Successful pipelines upload their results (artifacts) which can be browsed or accessed via the web API.
- **No special software setup required!**
- Artifacts containing root geometry can l[ink to a jsroot display of the MR's modified subsystem]