Geant-val Progress Report

- Alok Mathur
CERN EP-SFT

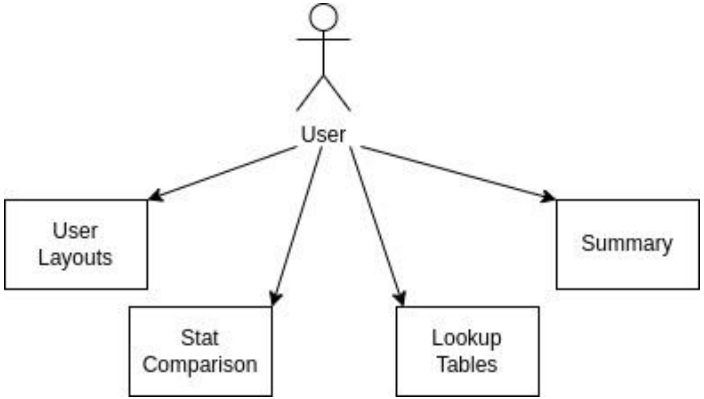
Introduction about Project

- I am working on developing a web application for Geant4 validation.
- The Geant-val web application serves as a centralized platform for visualizing and comparing results from community-developed tests run on Geant4's development releases, crucial for ensuring the reliability of Monte Carlo simulations in physics.
- Geant-val validates simulations by comparing them to experimental data, ensuring accuracy in modeling physical phenomena.

Software Components

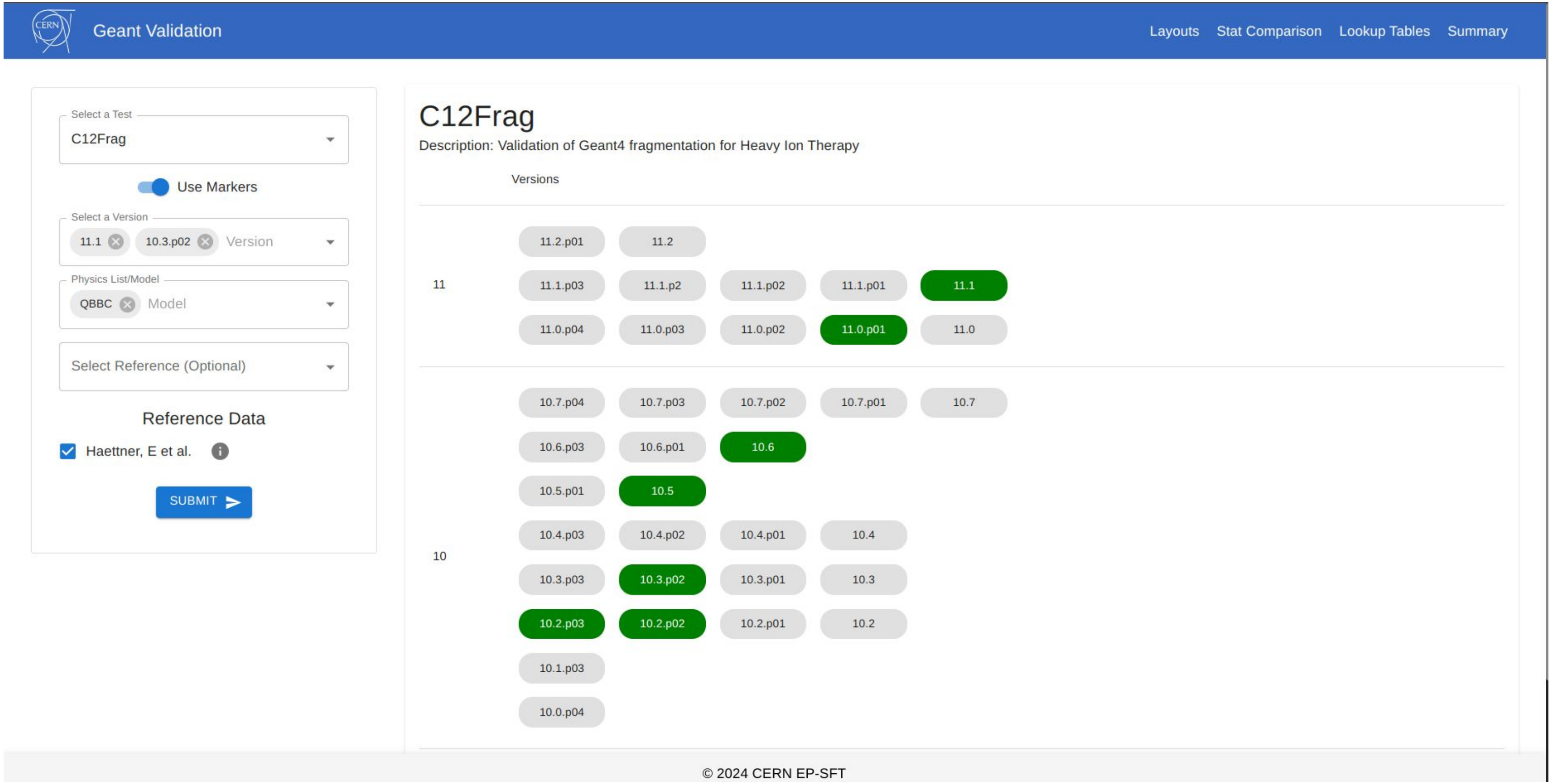
- The database is used for storing plots containing simulation results or experimental data, together with metadata describing these plots. PostgreSQL is used as database management system for the application. The database instance is provided by the CERN Database On-Demand service. The database schema is designed in a way to store scatter plots and histograms with unlimited number of optional test parameters in additional to a few mandatory ones.
- The server is the core of the Geant4 validation system. It provides a web API that allows clients to access the database, asynchronously responds to the clients requests and generates high quality plots "on the fly" whenever they are requested. The server is written in JavaScript and runs with the Node.js engine.
- The Web interface is an ReactJS single page application which shows plots with tests results together with statistical analysis

Modules in the Website



The Geant-val website provides two ways of viewing and comparing results:
Statistical comparisons page allows comparison of simulation with compatible experimental results using a selection of statistical test.
User Layouts It can be useful for Geant4 tests that produce hundreds of different plots, but for whose fast "visual" validation it is often enough to compare only a small well-defined subset of them.
Lookup Tables
The metadata associated with the database i.e. the available versions, models etc.
Summary
A section summarising the various tests and versions associated to them.

Module 1 - User Layout Section



Select a Test: C12Frag

Use Markers (on)

Select a Version: 11.1 ⊗ 10.3.p02 ⊗ Version

Physics List/Model: QBBC ⊗ Model

Select Reference (Optional)

Reference Data

☑ Haettner, E et al. ⓘ

SUBMIT

C12Frag

Description: Validation of Geant4 fragmentation for Heavy Ion Therapy

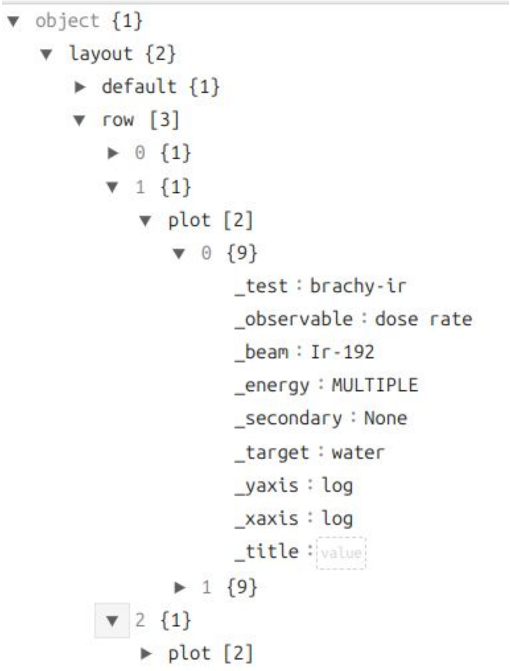
	Versions				
11	11.2.p01	11.2			
	11.1.p03	11.1.p2	11.1.p02	11.1.p01	🟩 11.1
	11.0.p04	11.0.p03	11.0.p02	🟩 11.0.p01	11.0
10	10.7.p04	10.7.p03	10.7.p02	10.7.p01	10.7
	10.6.p03	10.6.p01	🟩 10.6		
	10.5.p01	🟩 10.5			
	10.4.p03	10.4.p02	10.4.p01	10.4	
	10.3.p03	🟩 10.3.p02	10.3.p01	10.3	
	🟩 10.2.p03	🟩 10.2.p02	10.2.p01	10.2	
	10.1.p03				
	10.0.p04				

© 2024 CERN EP-SFT

Module 1 Ctd...

- The user selects the desired layout, Geant4 version(s), physics list(s) and experimental data.
- It allows performing fast visual comparison of several Geant4 versions/physics lists.
- Now after selecting the required options a JSON object is created and is used as an API for the ROOT C++ plotting utility.

```
{
    "selectedTestId": 129,
    "fileName": "brachy.xml",
    "selectedModels": [
        {
            "mctool_model_name": "emstandard_opt0"
        }
    ],
    "isMarkerSelected": true,
    "selectedVersions": [
        {
            "mctool_name_version_id": 348,
            "version": "11.2.p01"
        },
        {
            "mctool_name_version_id": 240,
            "version": "10.6"
        }
    ],
    "references": [
        {
            "expname": "D. Granero et al",
            "abstract": "A dosimetric study on the Ir-192 high dose rate
flexisource."
        }
    ]
}
```

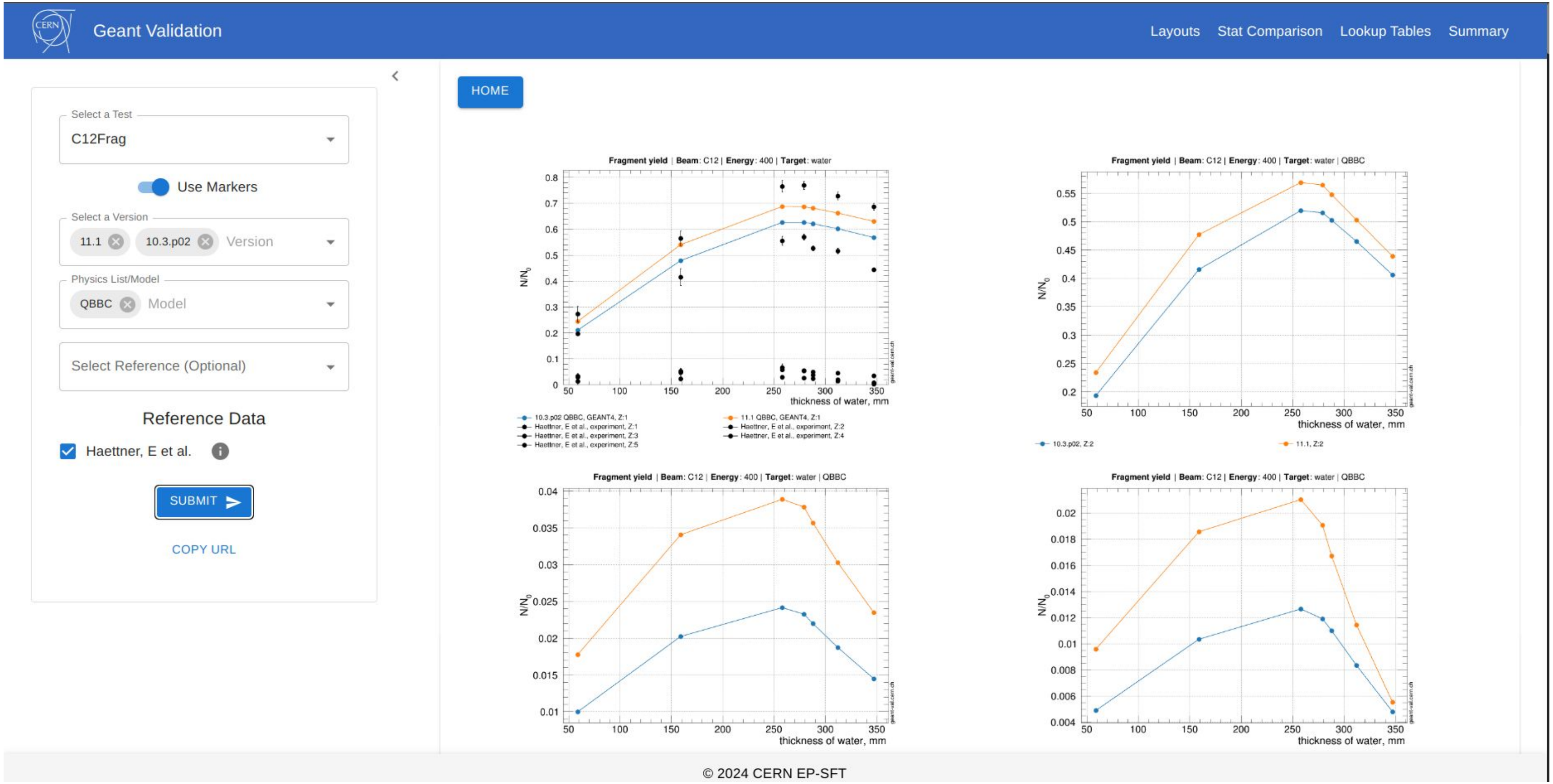


```
▼ object {1}
   ▼ layout {2}
      ▶ default {1}
      ▼ row [3]
         ▶ 0 {1}
         ▼ 1 {1}
            ▼ plot [2]
               ▼ 0 {9}
                     _test : brachy-ir
                     _observable : dose rate
                     _beam : Ir-192
                     _energy : MULTIPLE
                     _secondary : None
                     _target : water
                     _yaxis : log
                     _xaxis : log
                     _title : value
               ▶ 1 {9}
         ▼ 2 {1}
            ▶ plot [2]
```

Module 1 - Ctd...

- A ROOT-based C++ plotting utility was developed to produce high quality plots. It uses data in the JSON format which has been introduced as main interchange format between all parts of the application.
- It supports all types of application's data, can plot histograms with different binning on one canvas, and produce ratio plots. Ranges and scales of plot axes are selected automatically, but can be overridden if necessary.
- For plotting the JSROOT graphs, the ROOT binary file generated from plotting the above graph is used.
- npm package of JSROOT is used to plot the JSROOT graph.
- ROOT files along with the images generated are cached in the server side and are used if we get the same user input. So, that facilitates the faster retrieval of graphs. And also saves a lot of computation.

Module 1 - Ctd...


Geant Validation
Layouts
Stat Comparison
Lookup Tables
Summary
HOME
Fragment yield | Beam: C12 | Energy: 400 | Target: water
N/N$_0$
thickness of water, mm
10.3.p02 QBBC, GEANT4, Z:1
11.1 QBBC, GEANT4, Z:1
Haettner, E et al., experiment, Z:1
Haettner, E et al., experiment, Z:2
Haettner, E et al., experiment, Z:3
Haettner, E et al., experiment, Z:4
Haettner, E et al., experiment, Z:5
Fragment yield | Beam: C12 | Energy: 400 | Target: water | QBBC
10.3.p02, Z:2
11.1, Z:2
Fragment yield | Beam: C12 | Energy: 400 | Target: water | QBBC
Fragment yield | Beam: C12 | Energy: 400 | Target: water | QBBC
geant-val.cern.ch

Select a Test: C12Frag

Use Markers

Select a Version: 11.1 ⊗ 10.3.p02 ⊗ Version

Physics List/Model: QBBC ⊗ Model

Select Reference (Optional)

Reference Data

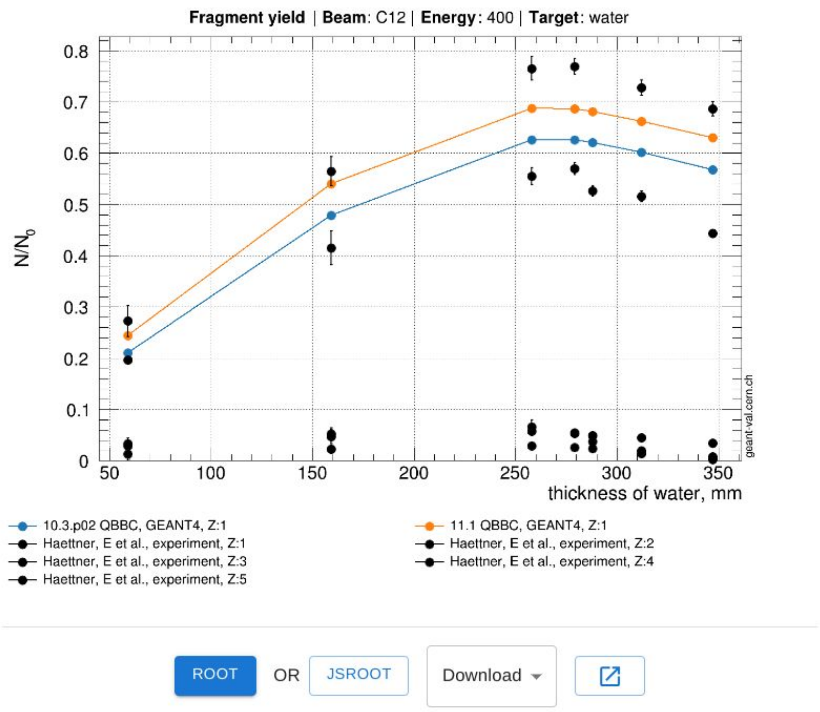
☑ Haettner, E et al.

SUBMIT

COPY URL

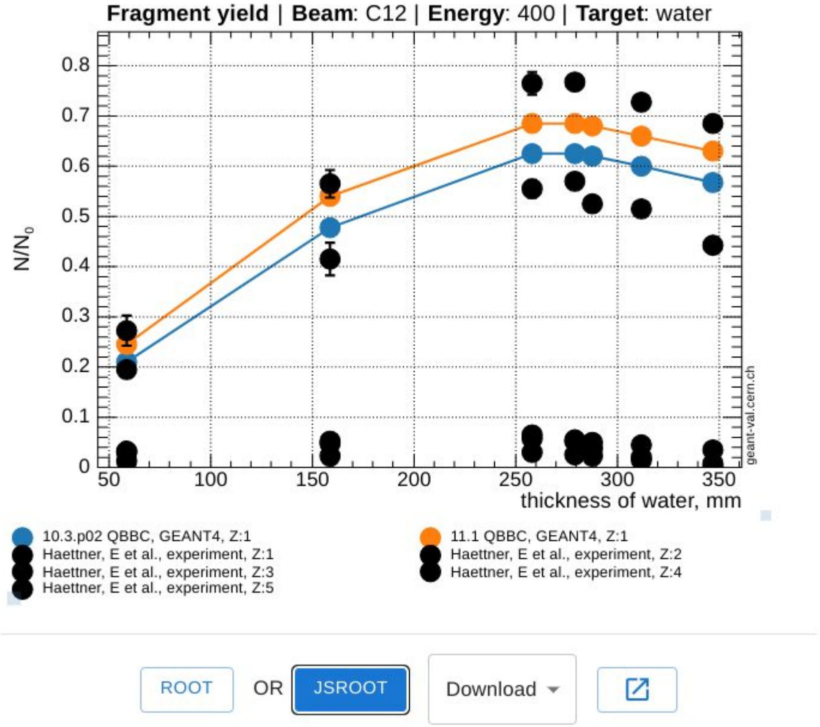
© 2024 CERN EP-SFT

Module 1 - Ctd...



Fragment yield | Beam: C12 | Energy: 400 | Target: water
N/N$_0$
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
50
100
150
200
250
300
350
thickness of water, mm
geant-val.cern.ch
10.3.p02 QBBC, GEANT4, Z:1
11.1 QBBC, GEANT4, Z:1
Haettner, E et al., experiment, Z:1
Haettner, E et al., experiment, Z:2
Haettner, E et al., experiment, Z:3
Haettner, E et al., experiment, Z:4
Haettner, E et al., experiment, Z:5
ROOT
OR
JSROOT
Download

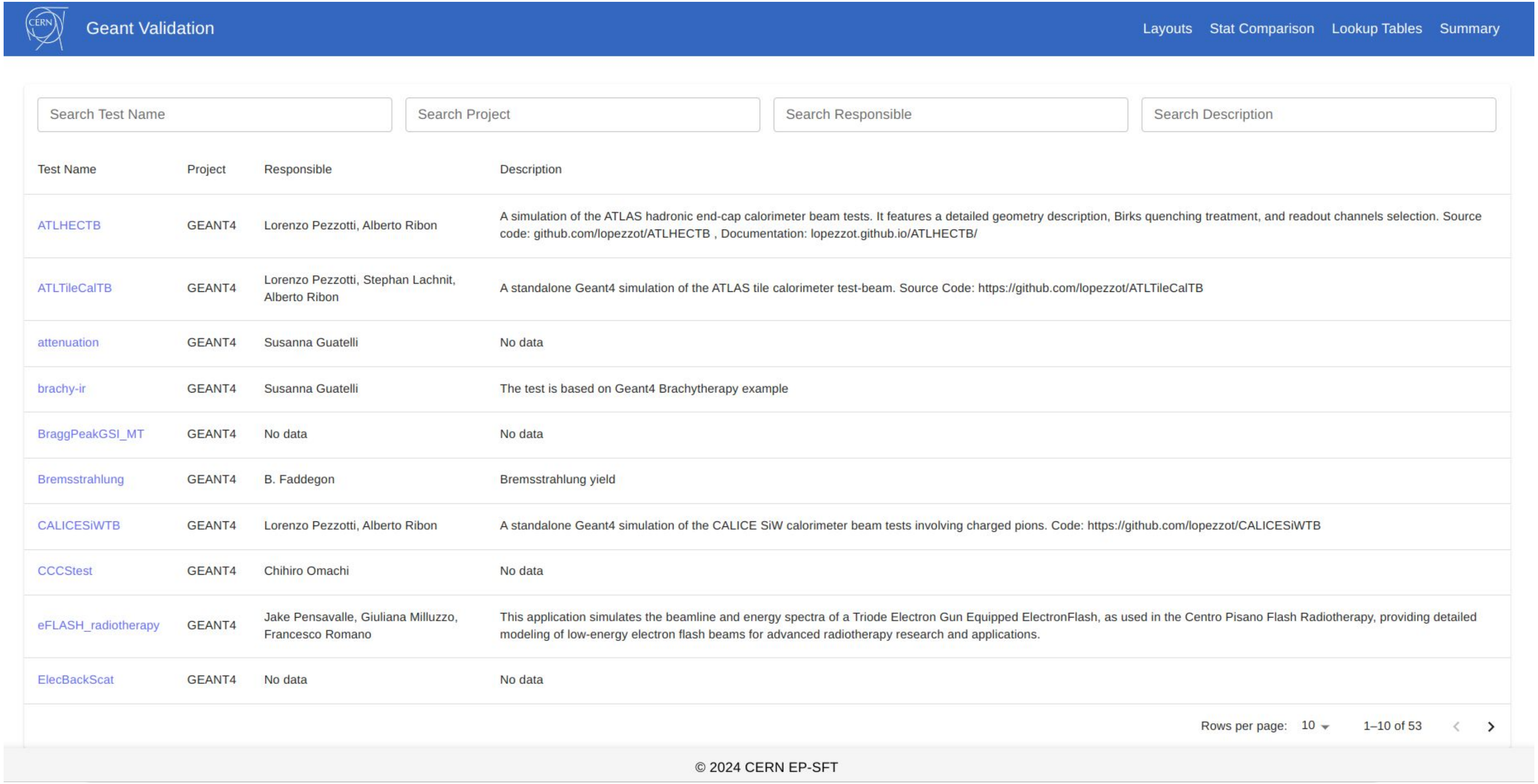
ROOT Plot



Fragment yield | Beam: C12 | Energy: 400 | Target: water
N/N$_0$
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
50
100
150
200
250
300
350
thickness of water, mm
geant-val.cern.ch
10.3.p02 QBBC, GEANT4, Z:1
11.1 QBBC, GEANT4, Z:1
Haettner, E et al., experiment, Z:1
Haettner, E et al., experiment, Z:2
Haettner, E et al., experiment, Z:3
Haettner, E et al., experiment, Z:4
Haettner, E et al., experiment, Z:5
ROOT
OR
JSROOT
Download

JSROOT Plot

Module 2 - Stat Comparison Section



Test Name	Project	Responsible	Description
ATLHECTB	GEANT4	Lorenzo Pezzotti, Alberto Ribon	A simulation of the ATLAS hadronic end-cap calorimeter beam tests. It features a detailed geometry description, Birks quenching treatment, and readout channels selection. Source code: github.com/lopezzot/ATLHECTB , Documentation: lopezzot.github.io/ATLHECTB/
ATLTileCalTB	GEANT4	Lorenzo Pezzotti, Stephan Lachnit, Alberto Ribon	A standalone Geant4 simulation of the ATLAS tile calorimeter test-beam. Source Code: https://github.com/lopezzot/ATLTileCalTB
attenuation	GEANT4	Susanna Guatelli	No data
brachy-ir	GEANT4	Susanna Guatelli	The test is based on Geant4 Brachytherapy example
BraggPeakGSI_MT	GEANT4	No data	No data
Bremsstrahlung	GEANT4	B. Faddegon	Bremsstrahlung yield
CALICESiWTB	GEANT4	Lorenzo Pezzotti, Alberto Ribon	A standalone Geant4 simulation of the CALICE SiW calorimeter beam tests involving charged pions. Code: https://github.com/lopezzot/CALICESiWTB
CCCStest	GEANT4	Chihiro Omachi	No data
eFLASH_radiotherapy	GEANT4	Jake Pensavalle, Giuliana Milluzzo, Francesco Romano	This application simulates the beamline and energy spectra of a Triode Electron Gun Equipped ElectronFlash, as used in the Centro Pisano Flash Radiotherapy, providing detailed modeling of low-energy electron flash beams for advanced radiotherapy research and applications.
ElecBackScat	GEANT4	No data	No data

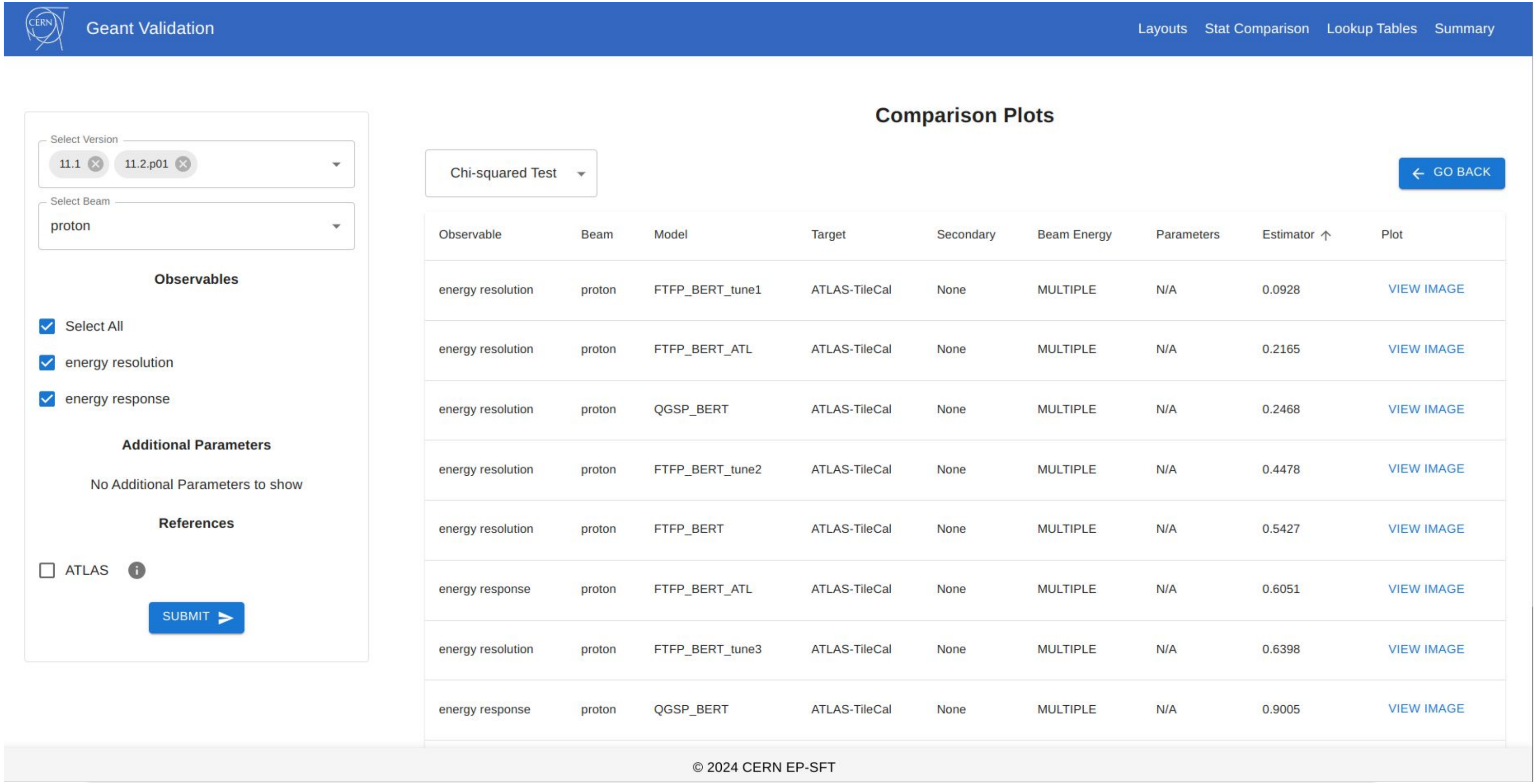
Rows per page: 10 1–10 of 53

© 2024 CERN EP-SFT

Module 2 - Ctd...

- Various tests and the associated metadata is shown and after the user selects certain tests it allows comparison of simulation with compatible experimental results using a selection of statistical tests.
- In this only 2 versions associated to a Geant4 test are used for comparison.
- The page shows results of Chi-squared test, Kolmogorov test and Maximal relative difference test.
- All computations are fast and performed asynchronously on the client side using JavaScript WebWorkers. For this purpose, JavaScript code to perform χ2 and Kolmogorov-Smirnov tests has been written

Module 2 - Ctd…



Select Version: 11.1 ⊗ 11.2.p01 ⊗

Select Beam: proton

Observables

☑ Select All

☑ energy resolution

☑ energy response

Additional Parameters

No Additional Parameters to show

References

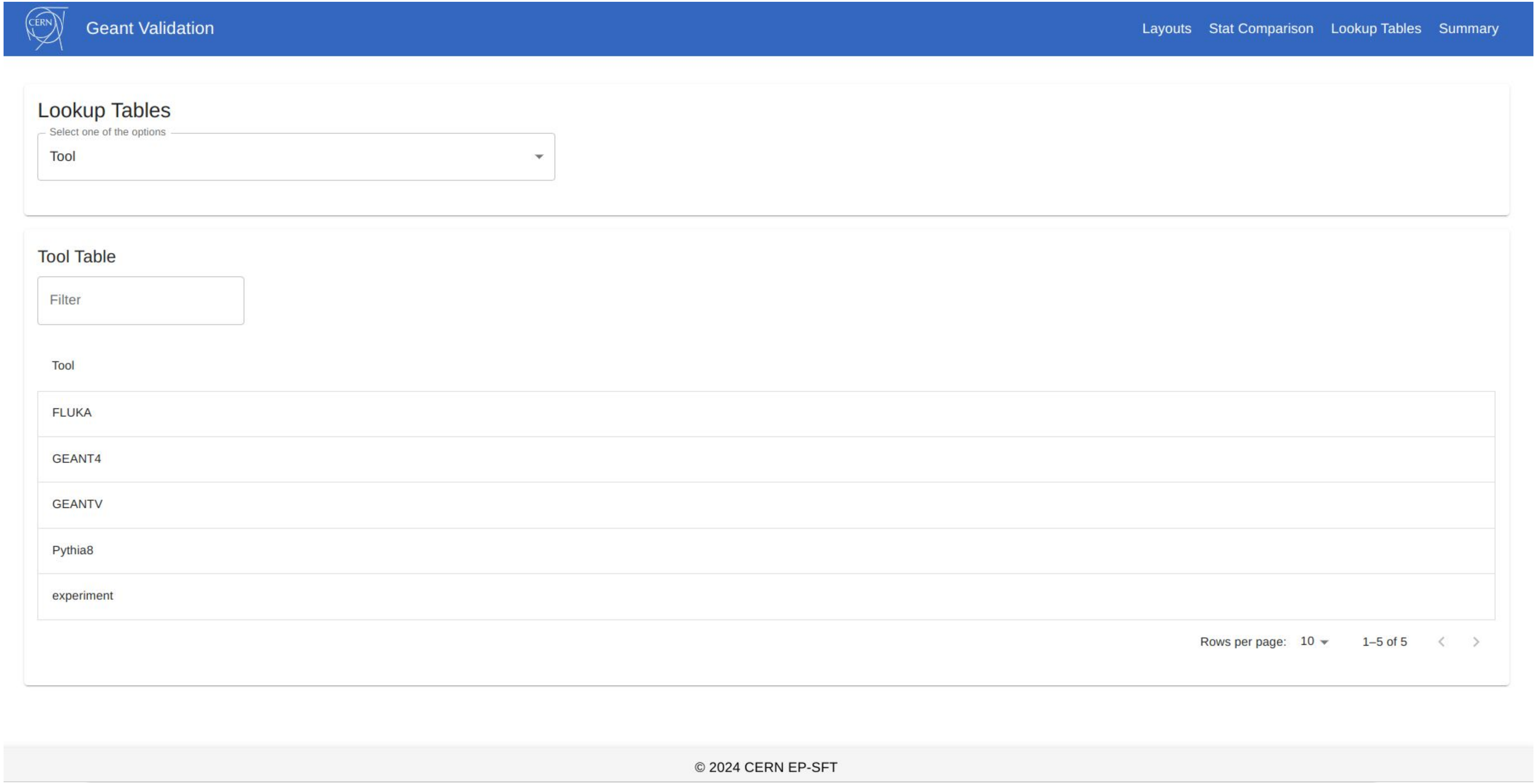
□ ATLAS

Comparison Plots

Chi-squared Test

Observable	Beam	Model	Target	Secondary	Beam Energy	Parameters	Estimator ↑	Plot
energy resolution	proton	FTFP_BERT_tune1	ATLAS-TileCal	None	MULTIPLE	N/A	0.0928	VIEW IMAGE
energy resolution	proton	FTFP_BERT_ATL	ATLAS-TileCal	None	MULTIPLE	N/A	0.2165	VIEW IMAGE
energy resolution	proton	QGSP_BERT	ATLAS-TileCal	None	MULTIPLE	N/A	0.2468	VIEW IMAGE
energy resolution	proton	FTFP_BERT_tune2	ATLAS-TileCal	None	MULTIPLE	N/A	0.4478	VIEW IMAGE
energy resolution	proton	FTFP_BERT	ATLAS-TileCal	None	MULTIPLE	N/A	0.5427	VIEW IMAGE
energy response	proton	FTFP_BERT_ATL	ATLAS-TileCal	None	MULTIPLE	N/A	0.6051	VIEW IMAGE
energy resolution	proton	FTFP_BERT_tune3	ATLAS-TileCal	None	MULTIPLE	N/A	0.6398	VIEW IMAGE
energy response	proton	QGSP_BERT	ATLAS-TileCal	None	MULTIPLE	N/A	0.9005	VIEW IMAGE

Module 3 - Lookup Tables Section



Lookup Tables

Select one of the options: Tool

Tool Table

Filter

Tool
FLUKA
GEANT4
GEANTV
Pythia8
experiment

Rows per page: 10 1–5 of 5

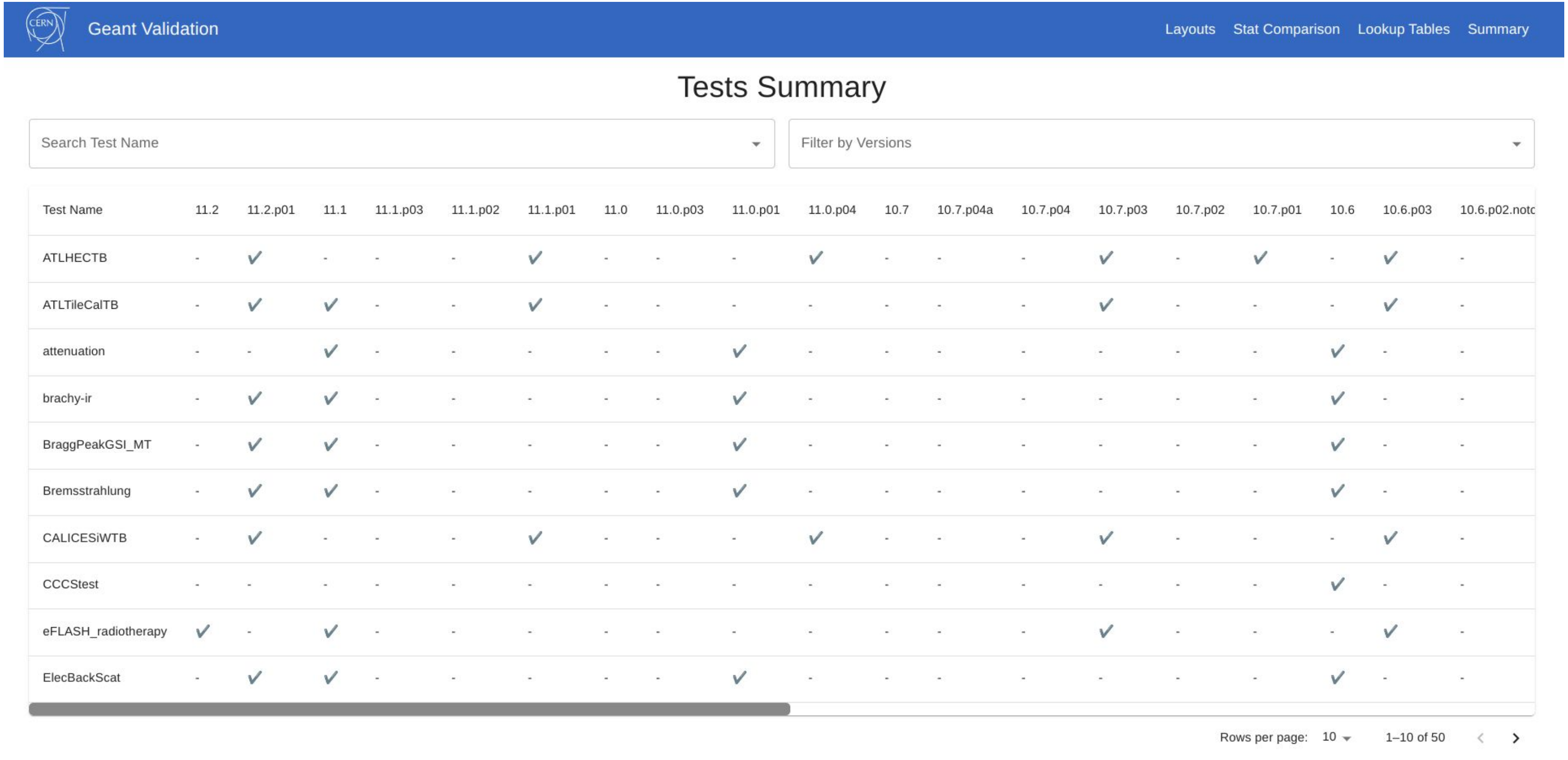
© 2024 CERN EP-SFT

Module 3 - Ctd...

The information stored in these tables are shown to user for reference.

- Tools
- Tests
- Observables
- Physics Model
- Versions
- Target
- Particles
- Articles

Module 4 - Summary Section



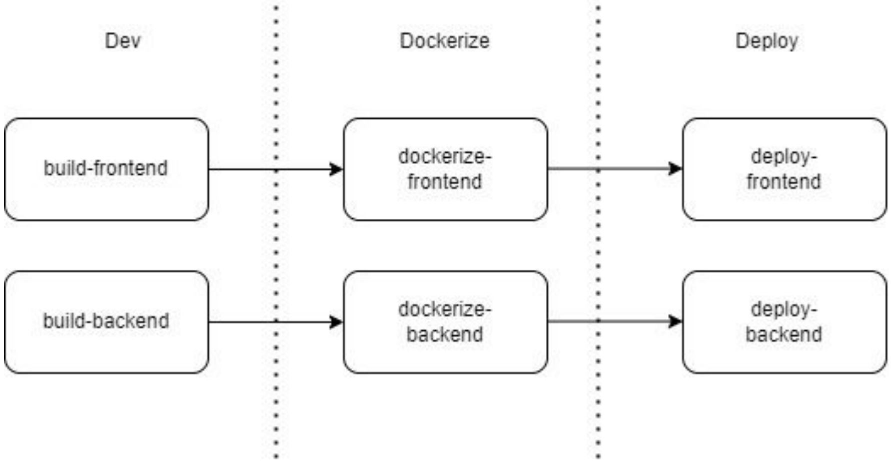
Tests Summary

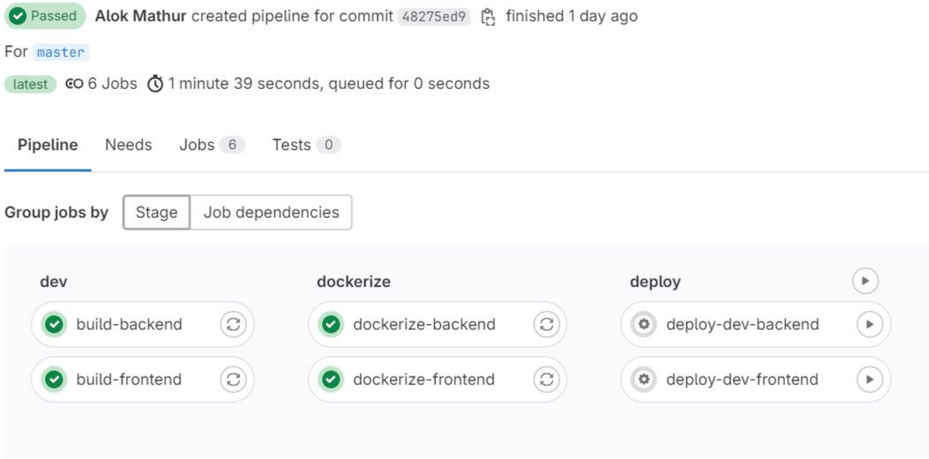
Test Name	11.2	11.2.p01	11.1	11.1.p03	11.1.p02	11.1.p01	11.0	11.0.p03	11.0.p01	11.0.p04	10.7	10.7.p04a	10.7.p04	10.7.p03	10.7.p02	10.7.p01	10.6	10.6.p03	10.6.p02.notc
ATLHECTB	-	✓	-	-	-	✓	-	-	-	✓	-	-	-	✓	-	✓	-	✓	-
ATLTileCalTB	-	✓	✓	-	-	✓	-	-	-	-	-	-	-	✓	-	-	-	✓	-
attenuation	-	-	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓	-	-
brachy-ir	-	✓	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓	-	-
BraggPeakGSI_MT	-	✓	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓	-	-
Bremsstrahlung	-	✓	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓	-	-
CALICESiWTB	-	✓	-	-	-	✓	-	-	-	✓	-	-	-	✓	-	-	-	✓	-
CCCStest	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-
eFLASH_radiotherapy	✓	-	✓	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	✓	-
ElecBackScat	-	✓	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓	-	-

Rows per page: 10 1–10 of 50

Deployment & CI/CD Pipeline

- For deployment the software used is OKD by Redhat for the deployment of the website, OKD, formerly known as OpenShift Origin, is an open-source distribution of Kubernetes, which is a popular container orchestration platform. Developed and maintained by Red Hat, OKD provides a platform for deploying, managing, and scaling containerized applications.
- The created CI/CD pipeline contains 3 stages and 6 jobs associated to it




Passed Alok Mathur created pipeline for commit 48275ed9 finished 1 day ago

For master

latest 6 Jobs 1 minute 39 seconds, queued for 0 seconds

Group jobs by Stage Job dependencies

dev
- build-backend
- build-frontend

dockerize
- dockerize-backend
- dockerize-frontend

deploy
- deploy-dev-backend
- deploy-dev-frontend

Deployment & CI/CD Pipeline

Stages Definition

- The pipeline is divided into three stages: dev, dockerize, and deploy. The dev stage is used for building the frontend and backend applications. The dockerize stage is responsible for creating Docker images for both the frontend and backend. The deploy stage is where the Docker images are deployed to the development environment.

Build Frontend

- The build-frontend job in the dev stage uses the node:18.17.1 image. It navigates to the gvp3-frontend directory, installs dependencies with npm install, and builds the frontend with npm run build. It caches the node_modules directory and stores the build output as artifacts.

Deployment & CI/CD Pipeline

Build Backend

- The job named build-backend is also part of the dev stage and uses the Node.js image node:14.17.0. The script for this job involves navigating to the gvp3-backend directory and installing the required dependencies with npm install. Similar to the frontend build, a cache is set up with the key *$CI_COMMIT_REF_NAME* to cache the *gvp3-backend/node_modules/* directory, speeding up subsequent builds.

Build Frontend Docker Image

- The dockerize-frontend job in the dockerize stage uses the gitlab-registry.cern.ch/ci-tools/docker-image-builder image. It depends on build-frontend, sets the image destination as ${CI_REGISTRY_IMAGE}/frontend:latest, and uses Kaniko(https://github.com/GoogleContainerTools/kaniko) to build and push the Docker image from gvp3-frontend/devops/Dockerfile.

Deployment & CI/CD Pipeline

Build Backend Docker Image

- The dockerize-backend job in the dockerize stage also uses gitlab-registry.cern.ch/ci-tools/docker-image-builder. It depends on build-backend, sets the image destination as ${CI_REGISTRY_IMAGE}/backend:latest, and uses Kaniko to build and push the Docker image from gvp3-backend/Dockerfile.

Deploy Frontend

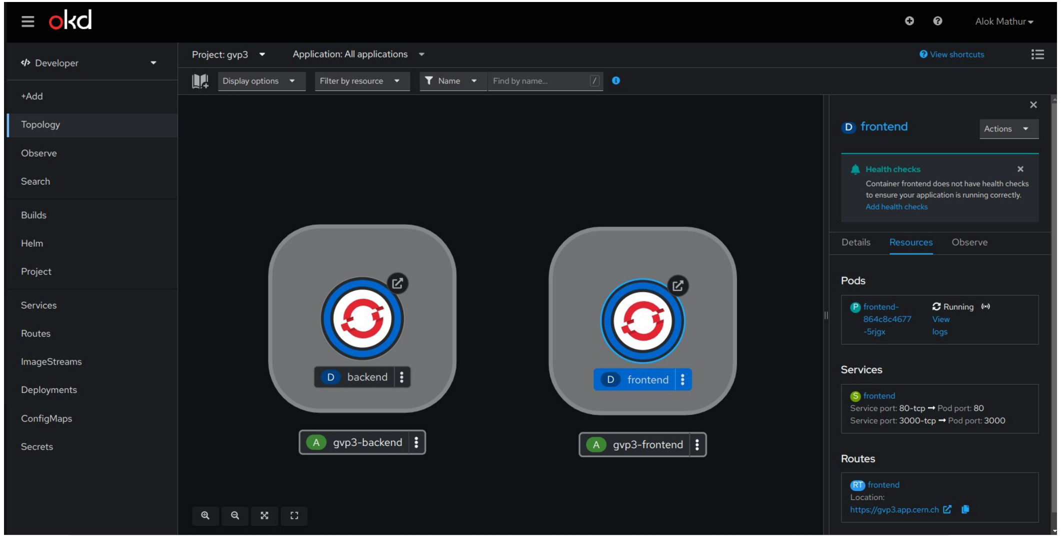
- The deploy-dev-frontend job in the deploy stage depends on dockerize-frontend. It uses the gitlab-registry.cern.ch/paas-tools/openshift-client:latest image. It imports the Docker image to OpenShift, waits for 15 seconds, and checks the rollout status. This job is manually triggered.

Deploy Backend

- The deploy-dev-backend job in the deploy stage depends on dockerize-backend. It uses the gitlab-registry.cern.ch/paas-tools/openshift-client:latest image. It imports the Docker image to OpenShift, waits for 15 seconds, and checks the rollout status. This job is also manually triggered.

Deployment & CI/CD Pipeline

```
kind: Deployment
apiVersion: apps/v1
metadata:
 annotations:
   alpha.image.policy.openshift.io/resolve-names: '*'
   app.openshift.io/route-disabled: 'false'
   deployment.kubernetes.io/revision: '44'
   image.openshift.io/triggers:
'[{"from":{"kind":"ImageStreamTag","name":"frontend:latest","namespace
":"gvp3"},"fieldPath":"spec.template.spec.containers[?(@.name==\"front
end\")].image","pause":"false"}]'
   openshift.io/generated-by: OpenShiftWebConsole
 resourceVersion: '2699056841'
 name: frontend
```



Health checks

Container frontend does not have health checks to ensure your application is running correctly.

Add health checks

Pods

P frontend-864c8c4677-5rjgx — Running — View logs

Services

S frontend
Service port: 80-tcp ➞ Pod port: 80
Service port: 3000-tcp ➞ Pod port: 3000

Routes

RT frontend
Location:
https://gvp3.app.cern.ch

Challenges Faced

Improper documentation for deploying a web app on OKD platform, the only documentation given is PAAS docs (https://paas.docs.cern.ch/) which wasn’t sufficient for deploying the JS web app for both Frontend and Backend.

Solution -> Creating a documentation of how to deploy a Web Application using the OKD platform.

Demonstration

Can view the website at cern.ch/gvp3

(or) can scan the below QR Code

Thank You