

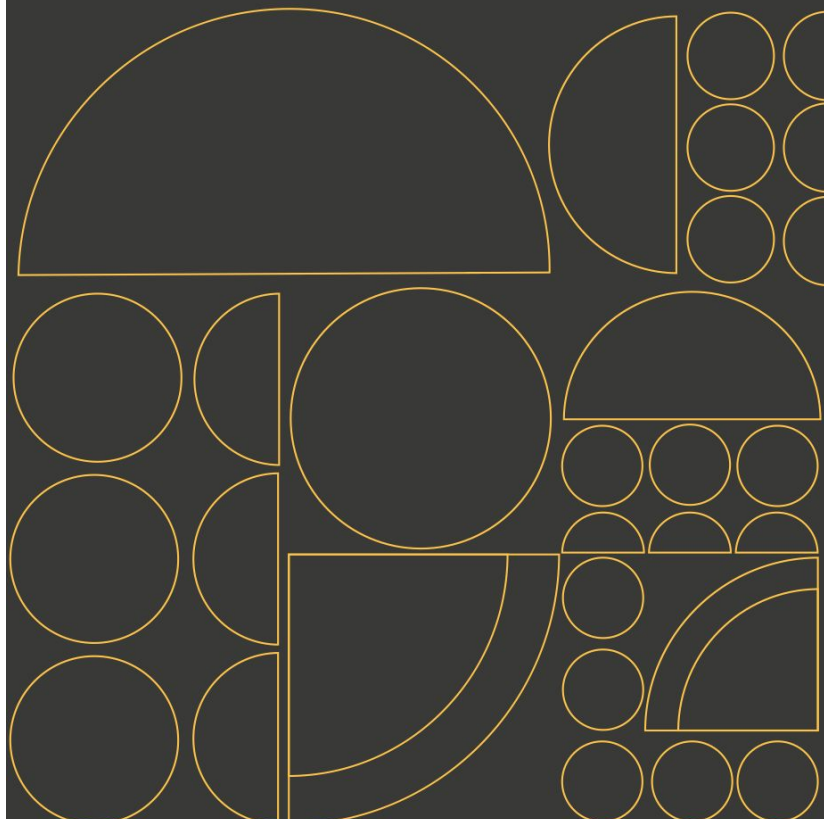
ChEESE

ChEESE CI-CD

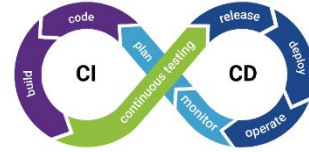
A.Folch, L.Mingari, H.Pascual
GEO3BCN CSIC



Project funded by EuroHPC under the grant agreement No 101093038.



What is CI/CD?



Formally: A methodology that automates the build, test, and deployment for the code

- **Faster** time to deploy
- Improves **code quality**
- Aids to the **team collaboration**
- **Reduces errors** and bugs



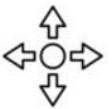
CODE



BUILD



TEST



DEPLOY

CASTIEL2 and ChESEE?



Codes in ChESEE-2P need to finish the CI/CD implementation before CASTIEL2 deploy its platform

CASTIEL is considering the next 3 options for the CI/CD platform:

Gitlab runners:

- Widely used by developers
- Can be installed in different locations
- Requires maintenance from developers
- Add some complexity in some cases as can be platform logins

EESI + CernVM:

European Environment for Scientific Software Installations

- Good HPC integration
- Requires deployment in HEs
- GPFS and Lustre filesystems not performing well
- Requires EasyBuild recipe and adds complexity to deploy

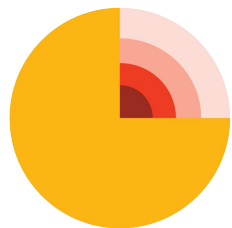
Jacamar:

- Based on GitLab runners
- Can launch jobs in HPC clusters bridging logins
- Deployment on HEs needed
- Reduced community

ChEESE CI/CD status



Code	Dev. platform	CI CD target	CI/CD methods	deployment methods	quality assessment	target systems in ChEESE-2P
SeiSol	GitHub		GitLab	Docker, Spack (instructions and files in repo of the SeiSol team)	unit-tests, end-to-end-tests, performance-regression-tests, clang-tools	x
SPECFEM3D	GitHub	Leonardo	travis, github	(undecided)	manual test cases	x
ExaHyPE	GitLab	Leonardo	GitLab	Dockerhub Spack	unit-tests, benchmarks (mini-apps), static code analysis (Python linter), clang-tools (clang-format)	x
Tandem	GitHub	Leonardo	travis, github	(container planed)	unit-tests, clang-tools	x
xSHELLS	bitbucket / gitlab	VEGA	GitLab	archive	manual test cases	VEGA, LUMI
HySEA	GitHub	VEGA	GitHub Actions	archive	global test case suite, run with "python test.py"	VEGA, Leonardo, MNS-BSC, LUMI
FALL3D	GitLab	Leonardo LUMI MNS	GitLab	Gitlab pipeline Docker executor	Linting, unit tests, coverage 50%, precision tests	Leonardo LUMI MNS
OpenPDAC	GitHub	LUMI-C	Github/actions LUMI (to be implemented)	local tests -> LUMI test & deploy	manual and/or automatic test cases	Leonardo LUMI
LaMEM	GitHub	LUMI-C	GitHub Actions	archive	testsuite (julia)	LUMI-C
PTatin3D	bitbucket	LUMI-C	x	archive	publication test cases	x
Elmer/Ice	Sourceforge	LUMI	GitHub/actions – LUMI	dep-package, Docker, Singularity, Spack	testsuite (ctest)	LUMI



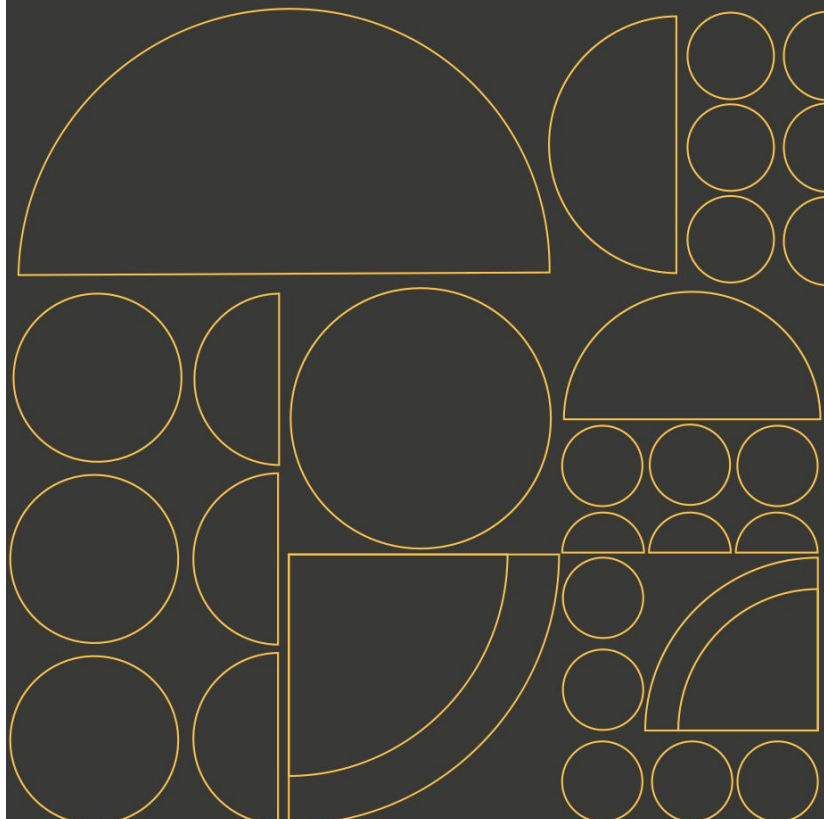
ChEESE

FALL3D CI-CD

A.Folch, L.Mingari, H.Pascual
GEO3BCN CSIC



Project funded by EuroHPC under the grant agreement No 101093038.





Introducing GitLab CI/CD

GitLab offers a complete DevOps platform, including CI/CD

- **Pipelines:** Define the overall workflow of your application
- **Stages:** Group related jobs into logical steps
- **Jobs:** Individual tasks within a pipeline
- **Runners:** Machines that execute jobs
- **Rules:** Conditions for job execution
- **Artifacts:** Output of jobs used in subsequent jobs
- **Secret Variables:** Sensitive variables for storing confidential information
- **Secret Files:** Files containing sensitive data.

The .gitlab-ci.yml file



Heart of the pipeline, the YAML file defines the entire CI/CD process.

Job structure:

- **Stages:** Different pipeline phases
 - build
 - test
 - deploy
- **Image:** Docker image to load
- **Rules:** Conditions for job execution
- **Jobs:** Specific tasks within each stage
- **Scripts:** Commands to be executed.

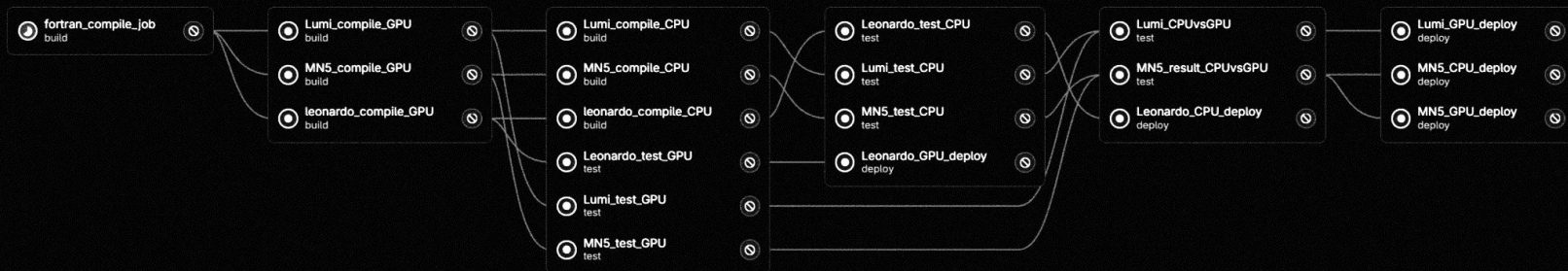
```
fortran_compile_job:
  stage: build
  image: albertogarcia/fortran-mpi
  only:
    changes:
      - Sources/**/*
      - Example/**/*
      - CI_CD/*
  before_script:
    - apt-get update && apt-get install -y cmake
  script:
    - mkdir MPI
    - cd MPI
    - cmake .. -DDETAIL_BIN=YES -DWITH-MPI=YES
    - make
```

FALL3D CI/CD Pipeline using Docker Executor



Docker Executor: A way for a GitLab runner to run jobs in Docker containers

- **Isolation:** Each job runs in its own container
- **Reproducibility:** Ensures a consistent environment
- **Scalability:** Easily add more runners
- **Parallelization:** Speeds up the process executing jobs simultaneously
- **Dependencies:** Jobs run when specific conditions are met (e.g. after successful build stage)



FALL3D CI/CD MN5 compile

```
MN5_compile_GPU:
  stage: build
  needs:
    - job: fortran_compile_job
      artifacts: false
  image: alpine
  rules:
    - changes:
      - "Sources/**/*"
      - "Example/**/*"
      - "CI_CD/mn5.yml"
  variables:
    GIT_STRATEGY: clone
    SECURE_FILES_DOWNLOAD_PATH: "/tmp/"
    LOGIN_NODE: "alogin1.bsc.es"
    TRANSFER_NODE: "transfer1.bsc.es/"
    TEST_DIR: "/gpfs/projects/bsc21/bsc412521/cicd"
    PRIVATE_KEY_PATH: "./tmp/cicd_id_rsa"
    MN5_USER: "bsc412xxx"
    BIN_NAME_ACC : "Fall3d.NVHPC.r8.mpi.acc.x"
    BIN_NAME_MPI : "Fall3d.NVHPC.r8.mpi.cpu.x"
    GPU4_PATH: "./tmp/MN5_4gpu_Tfall3d_sbatch.sh"
    CPU24_PATH: "./tmp/MN5_24cpu_Tfall3d_sbatch.sh"
```

```
before_script:
  - apk update > /dev/null 2>&1
  - apk add openssh curl git rsync bash > /dev/null 2>&1
  - curl --silent "https://secretsURL" | bash > /dev/null 2>&1
  - mv $PRIVATE_KEY_PATH /tmp/cicd_id_rsa
  - PRIVATE_KEY_PATH='./tmp/cicd_id_rsa'
  - mkdir -p Example/gpu
  - mkdir -p Example/cpu
  - mv $GPU4_PATH Example/gpu/MN5_4gpu_Tfall3d_sbatch.sh
  - mv $CPU24_PATH Example/cpu/MN5_24cpu_Tfall3d_sbatch.sh
  - rm -rf ./tmp
  - chmod 600 $PRIVATE_KEY_PATH
  - git checkout $CI_COMMIT_REF_NAME
script:
  - ssh -o StrictHostKeyChecking=no -o ConnectTimeout=5 -i
    $PRIVATE_KEY_PATH $MN5_USER@$LOGIN_NODE "ls 1>/dev/null"
  - ssh -o StrictHostKeyChecking=no -i $PRIVATE_KEY_PATH
    $MN5_USER@$LOGIN_NODE "rm -rf $TEST_DIR"
  - ssh -o StrictHostKeyChecking=no -i $PRIVATE_KEY_PATH
    $MN5_USER@$LOGIN_NODE "mkdir -p $TEST_DIR"
  - rsync -avzq -e "ssh -o StrictHostKeyChecking=no -i
    $PRIVATE_KEY_PATH" ./ $MN5_USER@$LOGIN_NODE:$TEST_DIR
  - ssh -o StrictHostKeyChecking=no
    -i $PRIVATE_KEY_PATH $MN5_USER@$LOGIN_NODE "
    module load nvidia-hpc-sdk/24.3 &&
    module load hdf5/1.14.1-2-nvidia-nvhpcx &&
    module load pnetcdf/1.12.3-nvidia-nvhpcx &&
    module load netcdf/c-4.9.2_fortran-4.6.1-nvidia-nvhpcx &&
    module load EB/apps &&
    module load CMake/3.27.6-GCCcore-13.2.0 &&
    mkdir -p $TEST_DIR/build_mpi_gpu &&
    cd $TEST_DIR/build_mpi_gpu &&
    cmake .. -DDETAIL_BIN=YES -DWITH-MPI=YES -DWITH-ACC=YES &&
    make -j 16 && "
```

FALL3D CI/CD Pipeline



ChEESE 24GA test

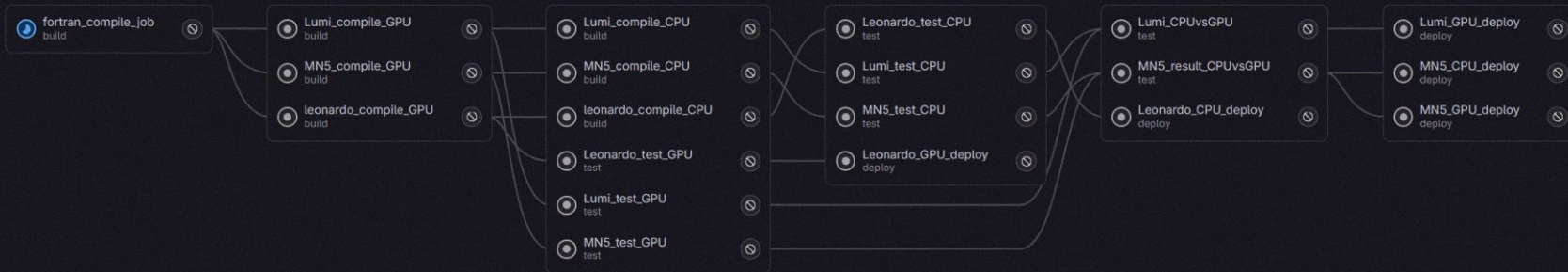
Running Heribert Pascual created pipeline for commit `a4db38d3` just now

For `cicd_multiplatform_local`

latest 60 20 jobs In progress, queued for 1 seconds

Pipeline Needs Jobs 20 Tests 0

Group jobs by Stage Job dependencies Show dependencies ☒



FALL3D CI/CD Failed



Lumi_compile_GPU

Failed Started 1 hour ago by Heribert Pascual

Search visible log output

```
1 Running with gitlab-runner 17.4.0-pre.110.g27408594 (27408594)
2   on blue-2.saas-linux-small-amd64.runners-manager.gitlab.com/default XxUrkriX, system ID: s_f46a98edce4
3   ✓ Preparing the "docker+machine" executor
4   Using Docker executor with image alpine ...
5   Pulling docker image alpine ...
6   Using docker image sha256:91ef0af61f39ece4d6718e465df5ed6ca12112358344fd51ae6a3b886634148b for alpine with digest alpine@sha256:beef
   dbd8a1da6d2915566fde36db9db0b524eb737fc57cd1367effd16dc0d06d ...
7   ✓ Preparing environment
8   Running on runner-xxurkriX-project-61170173-concurrent-0 via runner-xxurkriX-s-l-s-amd64-1726216413-57b3ad12...
9   ✓ Getting source from Git repository
10  Fetching changes with git depth set to 20...
11  Initialized empty Git repository in /builds/decube83/fall-3-d-multiplatform-cicd/.git/
12  Created fresh repository.
13  Checking out afdb38d3 as detached HEAD (ref is cicd_multiplatform_local)...
14  Skipping Git submodules setup
15  $ git remote set-url origin "${CI_REPOSITORY_URL}"
16  ✓ Executing "step_script" stage of the job script
17  Using docker image sha256:91ef0af61f39ece4d6718e465df5ed6ca12112358344fd51ae6a3b886634148b for alpine with digest alpine@sha256:beef
   dbd8a1da6d2915566fde36db9db0b524eb737fc57cd1367effd16dc0d06d ...
18  $ apk update > /dev/null 2>&1
19  $ apk add openssh curl bash > /dev/null 2>&1
20  $ echo -e "\e[102m    \e[0m LUMI dependencies installed"
21  [REDACTED] LUMI dependencies installed
22  $ curl -s -silent "https://gitlab.com/gitlab-org/incubation-engineering/mobile-devops/download-secure-files/-/raw/main/installer" | ba
   sh > /dev/null 2>&1
23  $ echo -e "\e[102m    \e[0m LUMI securefiles installed"
24  [REDACTED] LUMI securefiles installed
25  $ mv $PRIVATE_KEY_PATH /tmp/pr_id_rsa_lumi
26  $ PRIVATE_KEY_PATH="/tmp/pr_id_rsa_lumi"
27  $ chmod 600 $PRIVATE_KEY_PATH
28  $ git checkout $CI_COMMIT_REF_NAME
29  /bin/sh: eval: line 174: git: not found
30  ✓ Cleaning up project directory and file based variables
31  ERROR: Job failed: exit code 127
```

ChEESE 24GA test

Failed Heribert Pascual created pipeline for commit afdb38d3 1 hour ago, finished 1 hour ago

For `cicd_multiplatform_local`

latest 20 jobs 19.48 9 minutes 19 seconds, queued for 1 seconds

Pipeline Needs Jobs **20** **Failed Jobs 1** Tests 0

Name	Stage	Failure
Failed Lumi_compile_GPU	build	

```
$ echo -e "\e[102m    \e[0m LUMI securefiles installed"
LUMI securefiles installed
$ mv $PRIVATE_KEY_PATH /tmp/pr_id_rsa_lumi
$ PRIVATE_KEY_PATH="/tmp/pr_id_rsa_lumi"
$ chmod 600 $PRIVATE_KEY_PATH
$ git checkout $CI_COMMIT_REF_NAME
/bin/sh: eval: line 174: git: not found
Cleaning up project directory and file based variables
ERROR: Job failed: exit code 127
```

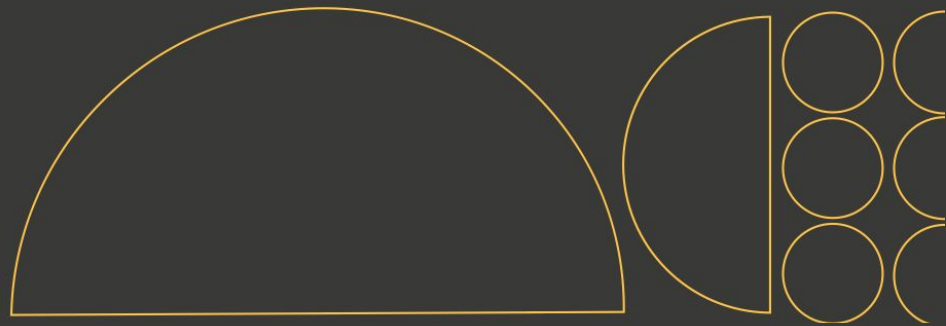
FALL3D CI/CD Problems found



During the development of the FALL3D's pipeline the team encountered several issues:

- **Pipeline design:** Where and when to test, which kind of tests, triggers...
- **Cluster accounts:** Necessity of code-specific accounts for CI/CD deployment
- **Authentication issues:** Difficulties in accessing Leonardo clusters via SSH (2FA)
- **CI/CD pipeline execution:** Defining triggers for pipeline runs (merge request, tag creation, ...)
- **Final deploy:** Determining the best method for software distribution (module, Spack)
- **HEs policies:** Concerns regarding security from automated deployments
- **HEs support:** Limited support for software sharing in some HEs

Thank you!



@cheese-coe



<http://cheese2.eu>



@cheese-coe

FALL3D CI/CD Pipeline



- **Build:** Into a container CPU, Leonardo CPU & GPU, MN5 CPU & GPU, Lumi GPU & CPU
- **Test:** Code execution in all platform CPU & GPU. Result comparison between them
- **Deploy:** Empty now, should create the module Fall3d for the target platforms

