NSBAS - gtilab-ci et HPC

Franck Thollard
et al.

ISTerre
Institut des Sciences de la Terre

Problématique

Tester la chaîne NSBAS et son environnement de travail/compilation

- tests de compilation sur plusieurs plateformes
- tests unitaires
- tests fonctionnels

La chaîne NSBAS

Traitement d'images satellites (IO et CPU)

- pas mal de dépendances : C, fortran, python, perl, hdf5, netcdf, fftw3f, blas/lapack, gdal, roi_pac
- pas mal d'utilisateurs sur Gricad (10-20)
- workflow complexe
- chaîne en constante évolution (outil de recherche)

Les utilisateurs...

Les utilisateurs et leurs besoins

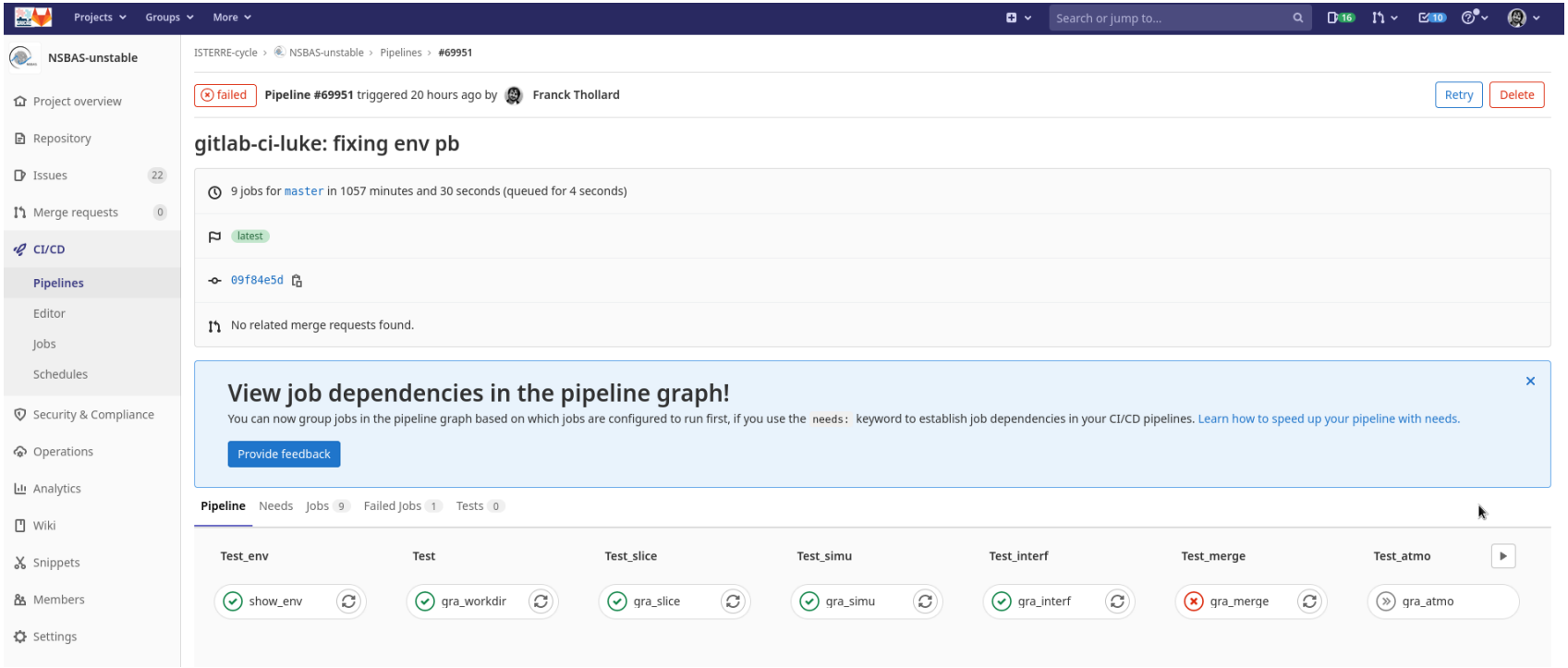
- core-dev $\rightarrow$ besoin d'avoir une instance propre de la chaîne que l'on peu bidouiller. Mais dev $\neq$ devops
- académiques locaux (e.g. Master2 geocsience, postdoc)
  $\rightarrow$ Env. prédéfini accessible facilement et qui fonctionne
- les partenaires scientifiques (Paris, Lyon, Strasbourg, ...) qui font tourner la chaîne chez eux et qui doivent la déployer
  $\rightarrow$ procédure de compilation/déploiement simple.
- contexte opérationnel : partenaires indus. (CNES), SNO, ... :
  $\rightarrow$ politique de gestion des versions et de tests.

NSBAS @ Gitlab-gricad

Code sur le gitlab de Gricad avec Intégration Continue.

- compilation sur runner partagé Gitlab (image debian)
- test sur un linux (pas sûr de maintenir)
- test sur gricad (runner partagé $\rightarrow$ VM ist-etalab $\rightarrow$ luke/dahu)
- test sur ist-oar (runner partagé $\rightarrow$ VM ist-etalab $\rightarrow$ ist-oar).

Workflow


Projects
Groups
More
Search or jump to...
NSBAS-unstable
ISTERRE-cycle › NSBAS-unstable › Pipelines › #69951
Project overview
Repository
Issues 22
Merge requests 0
CI/CD
Pipelines
Editor
Jobs
Schedules
Security & Compliance
Operations
Analytics
Wiki
Snippets
Members
Settings
failed Pipeline #69951 triggered 20 hours ago by Franck Thollard
Retry
Delete
gitlab-ci-luke: fixing env pb
9 jobs for master in 1057 minutes and 30 seconds (queued for 4 seconds)
latest
09f84e5d
No related merge requests found.
View job dependencies in the pipeline graph!
You can now group jobs in the pipeline graph based on which jobs are configured to run first, if you use the needs: keyword to establish job dependencies in your CI/CD pipelines. Learn how to speed up your pipeline with needs.
Provide feedback
Pipeline
Needs
Jobs 9
Failed Jobs 1
Tests 0
Test_env
Test
Test_slice
Test_simu
Test_interf
Test_merge
Test_atmo
show_env
gra_workdir
gra_slice
gra_simu
gra_interf
gra_merge
gra_atmo

Intégration continue : succès

- tests qui tournent par schedule : arrêté pour raison "eco-info". Réflexion en cours.
- choix des machines par tag : associer un tag à un runner, puis ajouter un tag à un job dans le .gitlab-ci.yml
- utilisation des variables pour accéder à la VM ist-etalab
- mise en place du workflow avec une granularité fine
- utilisation des variables pour choisir la partie du workflow qu'on teste.

Intégration continue : échecs

Tentative d'accès direct aux frontales de dahu/luke depuis les runners impossible
Tentative d'attaque de oar via api rest $\rightarrow$ échec

Example : test sur luke

setup

VM ist-etalab (shell executor) :

- ajout d'un service gitlab-runner
- ajout d'un user gitlab-runner
- connexion sans mot de passe de ist-etalab $\rightarrow$ {luke,dahu,ist-oar}
-

Gitlab-Ci - Schedule :

- définit la fréquence des runs (syntaxe crontab)
- définit ce que l'on test

Gitlab-Ci - Settings - CI/CD :

- Runner : ist-etalab : VM OSUG

Example : test sur luke

gitlab-ci.yml

- Variable GRICAD=gra-down-calc :
  - test sur Gricad sur le chantier "grenade" (**gra**)
  - des étapes de téléchargement (**down**) et réalise le calcul comple (**calc**)
- Variable cluster=luke : calcule sur luke, (alternative = dahu)

A venir

- Fusion des différents workflow en utilisant les variables
- Utilisation d'un git local pour récupérer un état particulier pour tester
- Continuous delivery : déploiement automatique si test ok.