



Guix

Contribuer à Guix-Science

Café Guix
27 janvier 2026

Romain Garbage
`romain.garbage@inria.fr`

Présentation de Guix-Science

Contribuer à Guix-Science

Contexte

- Utilisation de Guix pour le déploiement de logiciels scientifiques
- Sur sa/ses propres machines
- Sur des centres de calcul où Guix est installé (Abaca/Grid5000, GLiCID, GriCAD, PlaFRIM. . .)

Canaux Guix pour la science

Canaux "historiques"

- Guix HPC : <https://gitlab.inria.fr/guix-hpc/guix-hpc>
- Guix HPC non-free : <https://gitlab.inria.fr/guix-hpc/guix-hpc-non-free>

Canaux internes aux mésocentres

- GLiCID : <https://gitlab.univ-nantes.fr/glicid-public/guix-glicid>
- GriCAD : https://gricad-gitlab.univ-grenoble-alpes.fr/bouttief/gricad_guix_packages.git

Canaux liés à des PEPR

- PEPR Diadem :
<https://gricad-gitlab.univ-grenoble-alpes.fr/diamond/guix/guix-channel>
- PEPR NumPEX : <https://gitlab.inria.fr/numpex-pc5/guix-umpex>

Qu'est-ce que Guix-Science

- Projet Guix-Science : un ensemble de canaux pour étendre l'offre logicielle de Guix
- Créé par Lars-Dominik Braun, Ricardo Wurmus et d'autres personnes
- Initialement sur Github, migré sur Codeberg en octobre 2024.
- Canal <https://codeberg.org/guix-science/guix-science>
 - logiciels scientifiques en général
 - pas spécifique auHPC
- Canal <https://codeberg.org/guix-science/guix-science-non-free>
 - pour les logiciels propriétaires ou qui dépendent de briques propriétaires
- Canal <https://codeberg.org/guix-science/guix-past>
 - pour des logiciels du passé

Avantages de Guix-Science par rapport à Guix-HPC

- Forge ouverte (Codeberg)
- Contributions facilitées pour les personnes externes à l'INRIA
- Activité plus importante que Guix HPC
- Revue de code (Merci Simon et Ghislain en particulier) => Meilleure qualité du code
- Mise en place d'outils d'intégration continue

Pourquoi contribuer à Guix-Science

- Éviter la fragmentation de l'écosystème Guix
- Mutualiser les efforts d'empaquetage (ajouts ou maintenance de l'existant)
- Disposer d'un catalogue de logiciels scientifiques unifié dans les centres de calcul
- Moins intimidant que de contribuer dans Guix même
- Prendre de l'expérience pour contribuer à Guix même

Présentation de Guix-Science

Contribuer à Guix-Science

Premières étapes

- Créer un compte sur Codeberg : https://codeberg.org/user/sign_up
- Ajouter une clé SSH à son compte :
<https://docs.codeberg.org/security/ssh-key/>

Faire des modifications sur un clone en local

```
git clone https://codeberg.org/guix-science/guix-science
cd guix-science
$EDITOR
# Modifications locales...
guix build -L . foo # après avoir créé/modifié un paquet "foo"
guix lint -L . foo
# Vérification des modifications avec une révision plus récente de
# Guix sans avoir à mettre à jour l'environnement Guix global (le
# canal Guix principal est souvent modifié)
guix time-machine -q -- build -L . foo
guix time-machine -q -- lint -L . foo
```

Créer une requête de fusion : à partir d'un fork

Documentation

- <https://docs.codeberg.org/collaborating/pull-requests-and-git-flow/>

TL;DR

- Créer un `fork` de Guix-Science
- Cloner son fork en local
- Créer une branche dans son clone local
- Ajouter ses modifications au clone local
- Pousser la branche dans le dépôt distant (fork)
- Créer une requête de fusion à partir de l'interface web

Créer une requête de fusion : méthode AGit

- Documentation : <https://forgejo.org/docs/latest/user/agit-support/>

- Cloner le dépôt Guix-Science

```
git clone https://codeberg.org/guix-science/guix-science
```

- Créer une branche dans le dépôt local

```
git checkout -b ma_branche
```

- Ajouter ses modifications

```
git commit -a
```

- Créer une requête de fusion

```
git push origin HEAD:refs/for/main -o topic="ma_branche" \  
-o title="Titre" \  
-o description="Description"
```

Quelques conseils

- Suivre les Guix Guidelines : https://guix.gnu.org/manual/devel/en/html_node/Packaging-Guidelines.html
- Utiliser `guix lint` / `guix style`
- Utiliser les outils de CI mis en place :
 - Cuirass (<https://guix.bordeaux.inria.fr>)
 - Forgejo Actions (sur Codeberg)

CI : Cuirass et guix-science-bot

- Cuirass: outil de construction des paquets en continu (<https://codeberg.org/guix/cuirass/>)
- Instance dans la ferme de compilation Guix HPC hébergée à l'INRIA Bordeaux (<https://guix.bordeaux.inria.fr>)
- Chaque requête de fusion (PR) est évaluée et construite (<https://guix.bordeaux.inria.fr/pull-requests>)
- Les requêtes de fusion marquées **WIP** sont désactivées dans Cuirass
- @guix-science-bot : compte utilisé pour communiquer le résultat de l'évaluation de la PR
 - Paquets construits sans erreur => pastille verte
 - Évaluation ou construction de paquets échoués => pastille rouge

CI : Forgejo Actions

- `guix lint` sur les nouveaux paquets ou les paquets mis à jour
- TODO : `guix lint` sur tous les paquets modifiés ?
- TODO : `guix size` sur les nouveaux paquets/paquets mis à jour ?

Optionnel : signer ses commits

Documentation

- <https://docs.codeberg.org/security/gpg-key/#generating-a-gpg-keypair>

TL;DR

- Créer une clé GPG
- Exporter la clé publique
- Configurer Git pour signer ses commits
- Ajouter sa clé à Codeberg

Conclusion

- Projet communautaire
- Objectif : canal de référence pour les logiciels scientifiques
- Collaboration avec les centres de calculs utilisant Guix
- On attend vos contributions !