

ship

ship

“ Catalogue généré le 2026-05-11

En une phrase

Lance le workflow complet d'expédition de ton code : tests, revue de diff, bump de version, mise à jour du CHANGELOG, commit, push, et création de la Pull Request, tout automatiquement.

Quand l'utiliser

- Tu as fini une feature et tu veux la pousser sur GitHub en tant que PR prête à merger.
- Tu veux un workflow d'expédition strict (tests + revue + doc) avant chaque déploiement.
- Tu déploies plusieurs fois par jour et tu veux que tout soit automatisé et cohérent.
- Tu veux que la version (VERSION) et le CHANGELOG soient mis à jour proprement à chaque sortie.

Comment l'invoquer

- **Slash command** : `/ship`
- **Phrases déclencheurs (texte)** : "ship it", "create a pr", "push to main", "deploy this", "ship", "merge and push", "get it deployed"
- **Auto-invocation** : ☐ Oui — quand tu dis "le code est prêt", "on peut pousser", "crée une PR", Claude propose ce skill.

Description détaillée

Le skill `ship` est le workflow le plus ambitieux de gstack. Il enchaîne automatiquement, sans demander confirmation à chaque étape, toutes les opérations nécessaires pour transformer ton travail en cours en Pull Request prête à merger. La règle d'or : dire `/ship`, c'est dire "vas-y, fais tout". Le skill ne s'arrête que pour les vraies décisions humaines (bump de version majeur, conflits de merge, faille de revue critique).

Le workflow se décompose en une vingtaine d'étapes : préflight (vérifier que tu n'es pas sur la branche principale, lire `git status`), détection de la branche de base, exécution des tests, audit de couverture, vérification que les éléments du plan sont marqués DONE, revue avant atterrissage (pre-landing review qui détecte les bugs subtils), revue adversariale automatique (subagent Claude + challenge Codex), bump du VERSION avec choix automatique du niveau (PATCH/MICRO en auto, MINOR/MAJOR sur demande), génération de l'entrée CHANGELOG dans la voix produit, commit propre, push, et création de la PR avec le rapport complet en description.

C'est aussi un workflow idempotent : tu peux relancer `/ship` autant de fois que tu veux. Les vérifications tournent à chaque fois, mais les actions déjà faites (push, création de PR) ne sont pas dupliquées — la PR existante est mise à jour. Le skill produit à la fin l'URL de ta Pull Request. Si tu veux ensuite la merger et la déployer, tu enchaînes avec `/land-and-deploy`.

Source

- **Plugin** : `gstack`
- **Nom interne** : `ship`
- **Fichier** : `/home/thymon/.claude/skills/gstack/ship/SKILL.md`

Revision #2

Created 2026-05-11 21:18:57 UTC by thymon

Updated 2026-05-11 21:36:38 UTC by thymon