land-and-deploy

land-and-deploy

> *Catalogue généré le 2026-05-11*

En une phrase

Merge ta Pull Request, attend que le déploiement passe, puis vérifie automatiquement que la production tourne bien — l'étape qui suit `/ship`.

Quand l'utiliser

- Tu as fini `/ship` (ta PR est créée) et tu veux que tout le monde la merge et déploie sans toi.
- Tu veux que Claude surveille le déploiement et te dise si la production est saine après.
- Tu veux que la vérification post-déploiement (canary) tourne automatiquement.
- Tu déploies plusieurs fois par jour et tu veux que tout l'enchaînement soit géré pour toi.

Comment l'invoquer

- **Slash command** : `/land-and-deploy` (avec arguments : `/land-and-deploy #123`, `/land-and-deploy <url>`, `/land-and-deploy #123 <url>`)
- **Phrases déclencheurs (texte)** : "merge", "land", "deploy", "merge and verify", "land it", "ship it to production", "merge and deploy"
- **Auto-invocation** : □ Oui — Claude propose ce skill quand tu dis que c'est prêt à merger ou à déployer.

Description détaillée

Le skill `land-and-deploy` joue le rôle d'un ingénieur release qui a déployé des milliers de fois en production. Il sait que les deux pires moments en software sont les merges qui cassent la prod et les merges qui restent en file d'attente 45 minutes pendant que tu fixes l'écran. Sa mission : gérer ces deux moments gracieusement, en gardant ta confiance via des vérifications claires.

Le workflow démarre là où `/ship` s'arrête. `/ship` a créé la PR ; `/land-and-deploy` la merge, attend que la CI et le déploiement tournent, puis vérifie la santé de la production via des checks canary (un canari, des vérifications légères qui passent après chaque déploiement pour détecter les régressions). Sur ta première utilisation pour un projet, il fait un dry run : il détecte ton infrastructure de déploiement (Fly.io, Vercel, Netlify, Heroku, Railway), teste que ses commandes fonctionnent, et te montre exactement ce qui va se passer étape par étape avant de toucher quoi que ce soit. Une fois validé, il enregistre la config pour ne plus jamais reposer la question.

Sur les runs suivants, c'est en mode efficace : statut bref, peu d'explications. Le workflow est majoritairement automatisé. Il ne s'arrête que pour les vrais blocages : authentification GitHub manquante, pas de PR pour la branche, échecs de CI, conflits, échec de déploiement (avec proposition de revert), problèmes de santé détectés par le canary (avec proposition de revert). Le merge method (squash/merge/rebase) est auto-détecté depuis les settings du dépôt — pas de question pour ça non plus.

Source

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

Revision #2
Created 2026-05-11 21:19:17 UTC by thymon
Updated 2026-05-11 21:36:56 UTC by thymon