AI TOOLS WIKI · MÉMO

Claude Code Review *mémo*

Les commandes à connaître pour la revue de code dans **Claude Code** : la commande locale `/code-review`, la revue approfondie dans le cloud (« ultrareview ») et le service Code Review sur GitHub. Voir aussi la fiche **Claude Code Review**.

Famille Revue, sécurité et création de skills · **Éditeur** Anthropic · **Licence** Propriétaire
Note du wiki 11/15 · **Commandes vérifiées le** 25 septembre 2026
Fiche, notes et avis aitoolswiki.viellard.fr/outils/revue-et-securite/claude-code-review

1. Revue locale : `/code-review`

Intégrée à Claude Code (alias : `/review`). Elle tourne en arrière-plan dans un sous-agent : tu peux continuer à travailler, les résultats arrivent dans la conversation.

`/code-review`	Revoit les commits de ta branche non encore poussés + les changements non commités.
`/code-review 123`	Revoit la PR n° 123.
`/code-review ma-branche` ou `main...ma-feature`	Revoit une branche ou une plage de commits.
`/code-review src/auth.ts`	Revoit un fichier.
`/code-review --fix`	Applique les corrections dans le working tree après la revue.
`/code-review --comment`	Publie les résultats en commentaires inline sur la PR GitHub (ou une note sur une MR GitLab).
`/code-review low … max`	Choisit le niveau d'effort (voir ci-dessous).

Niveaux d'effort

Niveau	Comportement
`low`, `medium`	Seulement les problèmes dont la revue est très sûre : peu de faux positifs.
`high` à `max`	Couverture plus large, peut inclure des problèmes moins certains.
`ultra`	Revue multi-agents dans le cloud (voir section 2).

1. Revue locale : /code-review (suite)

Sans niveau précisé, la commande **réutilise le dernier niveau que tu as tapé**, même dans une session précédente.

2. Revue approfondie dans le cloud : `/code-review ultra`

`/code-review ultra`	Lance une flotte d'agents dans un bac à sable cloud, qui vérifient chaque problème indépendamment (5 à 10 min). Compare ta branche à la branche par défaut.
`/code-review ultra develop`	Compare à une autre branche de base.
`/code-review ultra vérifie mes changements d'auth`	Ajoute une consigne en texte libre à la revue.
`/code-review ultra --fix`	Applique les corrections à leur retour.
`/code-review ultra --post`	Sur une PR github.com : présélectionne la publication des résultats sur la PR.
`claude -p '/code-review ultra'`	Lance la revue cloud depuis un script ou une CI.

Coût : 3 revues offertes une seule fois sur Pro et Max, puis environ 5 à 25 $ par revue en crédits d'usage selon la taille du diff (estimation affichée avant chaque lancement). Nécessite un compte claude.ai ; indisponible sur Bedrock, Vertex, Foundry et en Zero Data Retention. Pour la CI qui doit attendre le résultat : `claude ultrareview [PR]` (options `--json`, `--timeout`, `--post`).

3. Service Code Review sur GitHub (Team et Enterprise)

Activé par un Owner dans les réglages admin de Claude Code. Les résultats arrivent en commentaires inline, classés 🔴 Important, 🟡 Nit, 🟣 Pré-existant. Le check « Claude Code Review » ne bloque jamais le merge.

Commentaire sur la PR	Effet
`@claude review`	Lance une revue unique.
`@claude review always`	Lance une revue et abonne la PR à une revue à chaque push.
`@claude review once`	Identique à `@claude review`.

À poster en **commentaire principal** de la PR (pas sur une ligne du diff), avec des droits d'écriture sur le dépôt. Coût moyen : 15 à 25 $ par revue.

Personnaliser la revue

Fichier	Rôle
`CLAUDE.md`	Règles générales du projet. Les écarts sont signalés en « Nit ».
`REVIEW.md` (racine du dépôt)	Règles propres à la revue : ce qui est « Important », nombre maximal de nits, chemins à ignorer, vérifications systématiques. **Lu par le service GitHub, pas par `/code-review` en local.**

Exemple de règles `REVIEW.md` : « Important = casse le comportement, fuite de données ou rollback bloqué » ; « au plus 5 nits par revue » ; « ignorer `src/gen/` et les `*.lock` » ; « toute nouvelle route d'API a un test d'intégration ».

Conseils d'utilisation

- **Relancer avant de pousser** : `/code-review` en local coûte peu et attrape les erreurs avant la PR.
- **Réserver `ultra` aux changements risqués** (auth, paiements, migrations) : c'est plus lent et payant.
- **Garder `REVIEW.md` court** : un fichier long dilue les règles qui comptent vraiment.
- **Limiter le bruit en re-revue** : ajouter dans `REVIEW.md` « après la première revue, ne signaler que les problèmes Important ».
- **Attention à `--fix` en arrière-plan** : `/rewind` ne l'annule pas. Utiliser git pour revenir en arrière.
- **Sur un dépôt très actif**, préférer le mode manuel (`@claude review`) au mode « à chaque push », qui multiplie les coûts.
- **Empêcher Claude de lancer la revue tout seul** : dans `settings.json`, `"skillOverrides": { "code-review": "user-invocable-only" }`.

Sources (documentation officielle)

- Documentation Code Review – Claude Code Docs
- Documentation Ultrareview – Claude Code Docs
- README du plugin code-review
- Niveaux d'effort de /code-review – ThakiCloud
- Guide /code-review et Ultrareview – Fastio