ETHANOLLE

CHECKLIST GRATUITE

Déployer Claude Code en entreprise sans risque

Les contrôles à parcourir pour autoriser Claude Code dans une équipe de développement : compte, confidentialité, code propriétaire, MCP, validation humaine et coûts.

7 contrôles

42 points à cocher

19 points essentiels

1 charte prête à diffuser

Ethanolle, partenaire certifié Anthropic

Édition 2026

Comment utiliser cette checklist

Cette checklist regroupe sept contrôles à réaliser pour autoriser Claude Code dans une équipe, vérifiés dans la documentation d'Anthropic et dans les tarifs de 2026.

Chaque point à cocher porte un niveau de priorité, afin que vous sachiez quoi traiter d'abord et quoi planifier une fois le **pilote** lancé.

ESSENTIEL	Point indispensable pour lancer un pilote, car il touche directement aux données de l'entreprise ou au contrôle du code produit.
RECOMMANDÉ	Point à traiter au cours du pilote, et à clôturer dès que l'usage s'étend à d'autres équipes que l'équipe pilote.
OPTIMISATION	Point qui renforce la maîtrise du déploiement sur la durée et vous aide à mesurer précisément les résultats concrets obtenus.

Vue d'ensemble des sept contrôles

N°	Contrôle	Points	Essentiels
1	Type de compte et mode d'accès	6	4
2	Paramètres de confidentialité et de rétention	6	3
3	Règles d'utilisation du code propriétaire	6	3
4	Accès aux outils internes et serveurs MCP	6	3
5	Validation humaine du code généré	6	3
6	Suivi de la consommation et des coûts	6	1
7	Règles à transmettre aux développeurs	6	2
	Total	**42**	**19**

Qui doit participer à la revue ?

Quatre profils devraient participer à la revue, car chacun détient une partie des réponses attendues par la checklist et utiles au déploiement.

- Responsable de la sécurité des systèmes d'information
- Délégué à la protection des données
- Responsable technique ou développeur référent
- Gestionnaire des licences, des achats ou de la finance

Une première lecture à plusieurs suffit généralement à repérer les points essentiels, sans qu'il soit nécessaire de configurer quoi que ce soit.

1 Type de compte et mode d'accès

Le type de compte détermine les règles de confidentialité, de rétention et d'administration qui s'appliquent au code de vos développeurs dès la première ligne générée.

Claude Code est inclus dans chaque abonnement Pro, Max, Team ou Enterprise de Claude, mais pas dans le plan gratuit, selon la page tarifs d'Anthropic.

- [ ] **Inventaire des comptes en usage** — ESSENTIEL
  Listez les comptes Claude réellement utilisés par vos développeurs, y compris les abonnements personnels **Pro ou Max** payés sur une carte individuelle.

- [ ] **Comptes personnels exclus du code d'entreprise** — ESSENTIEL
  Un compte **Pro ou Max** applique les réglages grand public : **cinq ans** de conservation si l'entraînement est autorisé, **trente jours** dans le cas contraire.

- [ ] **Offre professionnelle retenue** — ESSENTIEL
  Choisissez Team, Enterprise ou l'API : ces **conditions commerciales** excluent par défaut l'entraînement sur votre code, sauf programme choisi expressément par le client.

- [ ] **Connexion limitée à votre organisation** — ESSENTIEL
  Le réglage géré `forceLoginOrgUUID` épingle la connexion sur l'UUID de votre organisation, ce qui écarte les **comptes personnels** sur les postes de l'équipe.

- [ ] **Fournisseur d'accès comparé** — RECOMMANDÉ
  Comparez l'accès direct Anthropic, Amazon Bedrock, Google Cloud et Microsoft Foundry selon vos contrats cloud, votre hébergement et vos exigences de journalisation.

- [ ] **Surfaces autorisées** — RECOMMANDÉ
  Décidez quelles surfaces restent ouvertes : terminal, extensions VS Code et JetBrains, application desktop, sessions cloud et Remote Control, activables par les Owners Team ou Enterprise.

> **PIÈGE FRÉQUENT**
> Un développeur qui teste Claude Code avec son abonnement personnel envoie du code d'entreprise sous des règles de rétention grand public, sans décision de l'entreprise.

Responsable : Échéance : Notes :

2 Paramètres de confidentialité et de rétention

La confidentialité repose sur trois leviers : le contrat applicable, les réglages techniques et la durée de conservation des transcriptions, localement comme chez Anthropic.

Les points ci-dessous reprennent la documentation officielle d'Anthropic sur l'usage des données, consultée en 2026, avec les durées exactes publiées.

- [ ] **Conditions commerciales confirmées** — ESSENTIEL
  Sous les conditions Team, Enterprise ou API, Anthropic n'entraîne pas ses modèles sur le code ni sur les prompts, sauf programme choisi par le client.

- [ ] **Durée de rétention connue** — ESSENTIEL
  La conservation standard sous **conditions commerciales** dure **trente jours**, et la **rétention zéro** reste réservée aux comptes **Claude for Enterprise** qualifiés, activée organisation par organisation.

- [ ] **Revue par le délégué à la protection des données** — ESSENTIEL
  Soumettez à votre **DPO** le contrat applicable, les flux de données, les transferts hors Union européenne et le besoin éventuel d'une **analyse d'impact**.

- [ ] **Transcriptions locales maîtrisées** — RECOMMANDÉ
  Claude Code conserve les transcriptions de session sur le poste, en clair, pour **trente jours** par défaut, ce qui impose un **disque chiffré**.

- [ ] **Télémétrie décidée** — RECOMMANDÉ
  Claude Code envoie une télémétrie opérationnelle par défaut sur les sessions Anthropic, que la variable `DISABLE_TELEMETRY` coupe, au détriment du **tableau de bord** d'analyse.

- [ ] **Retours de session encadrés** — RECOMMANDÉ
  Les transcriptions partagées par la commande de retour sont conservées **cinq ans**, ce qui justifie une consigne claire donnée aux développeurs.

> **PIÈGE FRÉQUENT**
> La **rétention zéro** se confond souvent avec le réglage standard, alors qu'elle exige un accord spécifique avec Anthropic pour un compte qualifié.

Responsable : Échéance : Notes :

3 Règles d'utilisation du code propriétaire

Tout le code d'une entreprise ne présente pas le même niveau de sensibilité, et la règle d'usage doit refléter cette différence dépôt par dépôt.

Une règle écrite et appliquée techniquement protège mieux vos actifs qu'une simple recommandation envoyée par courriel à l'ensemble des développeurs.

- [ ] **Classification des dépôts** — ESSENTIEL
  Classez chaque dépôt selon quatre niveaux, public, interne, sensible et secret, puis associez à chaque niveau une décision claire : autorisé, restreint ou interdit.

- [ ] **Fichiers sensibles bloqués** — ESSENTIEL
  Une règle `permissions.deny` peut interdire la lecture de fichiers comme `.env` ou du dossier **secrets**, exemple que la documentation officielle donne pour sa **politique gérée**.

- [ ] **Secrets hors des prompts** — ESSENTIEL
  Aucune clé, aucun mot de passe ni aucun jeton ne figure dans un prompt, un fichier de contexte ou un jeu de test pour l'agent.

- [ ] **Dossiers et réseau limités** — RECOMMANDÉ
  Les réglages `additionalDirectories` et sandbox encadrent les dossiers accessibles et les domaines joignables, avec un verrou géré pour n'honorer que les listes de l'administrateur.

- [ ] **Données de production exclues** — RECOMMANDÉ
  Les jeux de test et les extraits de journaux fournis à l'agent ne contiennent aucune donnée personnelle réelle, ce qui limite le périmètre du traitement.

- [ ] **Propriété et licences du code généré** — OPTIMISATION
  Le service juridique valide la clause contractuelle sur la propriété du code généré et la politique de contrôle des licences open source des dépendances proposées.

> **PIÈGE FRÉQUENT**
> Une règle de refus écrite dans les réglages personnels reste modifiable par le développeur, alors qu'une **règle gérée** par l'organisation prime sur tous les niveaux.

Responsable : Échéance : Notes :

4 Accès aux outils internes et serveurs MCP

Un serveur MCP, pour Model Context Protocol, donne à l'agent un accès direct à vos outils internes, bien au-delà du dépôt de code.

Les clés citées ci-dessous proviennent de la documentation officielle sur les réglages gérés, consultée en 2026, et se déploient au niveau de l'organisation.

- [ ] **Inventaire des serveurs demandés** — ESSENTIEL
  Recensez les **serveurs MCP** que les équipes souhaitent connecter, avec pour chacun le propriétaire, les données exposées et les actions possibles.

- [ ] **Liste blanche réellement active** — ESSENTIEL
  Le verrou `allowManagedMcpServersOnly` n'a d'effet que si une **liste** `allowedMcpServers` existe : sans liste, tous les serveurs qui passent la liste noire se chargent.

- [ ] **Droits du compte connecté** — ESSENTIEL
  Un serveur MCP agit avec les droits du compte qui l'alimente : préférez un compte de service en lecture seule, limité aux seules ressources nécessaires.

- [ ] **Liste noire et serveurs fournis** — RECOMMANDÉ
  La clé `deniedMcpServers` bloque des serveurs précis, tandis que `managedMcpServers` ou `managed-mcp.json` fournit à chaque développeur les serveurs validés par l'organisation.

- [ ] **Plugins, hooks et extensions cadrés** — RECOMMANDÉ
  Les clés `strictKnownMarketplaces`, `strictPluginOnlyCustomization` et `allowManagedHooksOnly` réservent à l'administrateur le choix des plugins, des hooks et des sources d'extension autorisés.

- [ ] **Revue périodique des accès** — OPTIMISATION
  Planifiez une revue trimestrielle des serveurs actifs et des droits associés, car les outils internes évoluent plus vite que la politique qui les encadre.

> **PIÈGE FRÉQUENT**
> Un serveur MCP connecté avec un jeton d'administrateur donne à l'agent les mêmes droits que cet administrateur, sur chaque service que le serveur atteint.

Responsable : Échéance : Notes :

5 Validation humaine du code généré

Un agent produit du code rapidement, mais la responsabilité de ce qui entre en production reste humaine et se vérifie à des étapes précises.

Selon l'enquête Stack Overflow d'avril 2026, **63 %** des professionnels laissent rarement ou jamais un agent travailler seul, une prudence que ces contrôles formalisent.

- [ ] **Revue obligatoire à chaque fusion** — ESSENTIEL
  Aucun code généré n'entre dans la branche principale sans **pull request** relue par un développeur humain, différent de la personne qui a formulé la demande.

- [ ] **Mode de contournement désactivé** — ESSENTIEL
  Le réglage `permissions.disableBypassPermissionsMode`, posé au niveau géré, retire aux développeurs la possibilité de supprimer les demandes d'autorisation de Claude Code.

- [ ] **Tests et contrôles automatiques** — ESSENTIEL
  L'intégration continue exécute les tests, l'analyse statique et la détection de **secrets** sur chaque modification, quelle que soit la part de code écrite par l'agent.

- [ ] **Permissions gérées uniquement** — RECOMMANDÉ
  Avec `allowManagedPermissionRulesOnly`, seules les règles de permission de la politique centrale s'appliquent, et les règles locales, de projet ou de ligne de commande sont ignorées.

- [ ] **Relecture renforcée des zones critiques** — RECOMMANDÉ
  Authentification, paiement, chiffrement, migrations de base de données et droits d'accès exigent un second relecteur, y compris pour un changement de quelques lignes.

- [ ] **Traçabilité des commits** — OPTIMISATION
  Une règle d'équipe précise si les commits mentionnent l'assistance de Claude, que la clé attribution des réglages permet de conserver ou de retirer.

> **PIÈGE FRÉQUENT**
> Un diff volumineux relu rapidement laisse passer des erreurs : selon Stack Overflow 2025, **66 %** des personnes interrogées citent des solutions presque justes comme première frustration.

Responsable : Échéance : Notes :

6 Suivi de la consommation et des coûts

Une facturation à l'usage ou des limites partagées peuvent surprendre une équipe, d'où l'intérêt de fixer un cadre budgétaire dès le **pilote**.

Les mécanismes ci-dessous figurent dans la documentation officielle sur les coûts de Claude Code, consultée en 2026, et dans la page tarifs d'Anthropic.

- [ ] **Modèle de facturation choisi** — ESSENTIEL
  Abonnement Team ou Enterprise, ou compte API facturé au token : ce choix détermine qui contrôle la dépense et comment elle se plafonne.

- [ ] **Limites de dépenses posées sur l'API** — RECOMMANDÉ
  Sur l'API, les limites de dépenses se fixent par espace de travail, et Claude Code crée un espace dédié lors de la première connexion.

- [ ] **Limites par groupe ou par membre** — RECOMMANDÉ
  Avec un abonnement, les crédits d'usage et leurs limites se règlent au niveau de l'organisation, d'un groupe ou d'un membre, selon la documentation.

- [ ] **Limite de débit protégée** — RECOMMANDÉ
  Une limite de débit propre à l'espace de travail Claude Code protège vos autres charges de production, car son trafic compte dans les limites globales.

- [ ] **Tableau de bord et export branchés** — RECOMMANDÉ
  Le tableau d'analyse de Claude Code, l'API d'analyse par utilisateur et l'export **OpenTelemetry** alimentent vos propres outils de suivi de la consommation.

- [ ] **Revue mensuelle du coût par tâche** — OPTIMISATION
  Comparez chaque mois le coût par développeur et par type de tâche avec le temps économisé, afin de décider quels sièges Standard ou Premium conserver.

> **PIÈGE FRÉQUENT**
> Les limites d'un abonnement se partagent entre le chat et Claude Code, si bien qu'un usage intensif de l'un réduit la capacité disponible pour l'autre.

Responsable : Échéance : Notes :

7 Règles à transmettre aux développeurs

Les meilleures configurations échouent si les développeurs ignorent les règles, d'où l'importance d'une **charte** courte, d'un référent identifié et d'un **pilote** cadré.

Le modèle de **charte** fourni en fin de document peut servir de base, après adaptation à votre contexte et à votre politique de sécurité.

- [ ] **Charte d'usage diffusée** — ESSENTIEL
  Une **charte** d'une page, signée par chaque développeur, résume usages autorisés, interdits et réflexes de revue, sur le modèle fourni en fin de document.

- [ ] **Procédure en cas d'incident** — ESSENTIEL
  Une procédure écrite indique quoi faire si un secret apparaît dans un prompt : révocation immédiate, rotation, vérification des journaux et information du responsable sécurité.

- [ ] **Fichier `CLAUDE.md` partagé** — RECOMMANDÉ
  Un fichier `CLAUDE.md` versionné dans chaque dépôt consigne les conventions, les commandes de build et les interdits que l'agent doit respecter à chaque session.

- [ ] **Version minimale imposée** — RECOMMANDÉ
  Le réglage `requiredMinimumVersion` empêche le démarrage d'une version obsolète, mais il n'interrompt pas une session déjà ouverte depuis plusieurs semaines.

- [ ] **Pilote cadré** — RECOMMANDÉ
  Le **pilote** réunit une équipe restreinte, une durée définie, des critères de succès chiffrés et une revue formelle des incidents, des coûts et des retours.

- [ ] **Référent et canal d'aide** — OPTIMISATION
  Un référent nommé et un canal de discussion dédié absorbent les questions des premières semaines, ce qui évite que chacun invente ses propres pratiques.

> **PIÈGE FRÉQUENT**
> Une politique stricte déployée sans explication pousse les développeurs vers des outils non encadrés, ce qui déplace le risque au lieu de le réduire.

Responsable : Échéance : Notes :

Modèle de charte d'usage pour les développeurs

Ce modèle d'une page peut être adapté à votre contexte, puis diffusé aux développeurs au moment de l'ouverture des accès à Claude Code.

1. Seuls les comptes professionnels fournis par l'entreprise servent à utiliser Claude Code, et aucun abonnement personnel ne traite du code qui appartient à l'entreprise.
2. Aucun secret, mot de passe, jeton ou donnée personnelle réelle n'est copié dans un prompt, un fichier de contexte ou un jeu de test.
3. Les dépôts classés sensibles ou secrets restent exclus de l'usage de l'agent, sauf autorisation écrite accordée par le responsable de la sécurité.
4. Seuls les serveurs MCP et les plugins validés par l'organisation sont connectés, sans contournement des réglages gérés par l'administrateur de l'entreprise.
5. Le mode de contournement des permissions reste désactivé, et chaque action sensible demandée par l'agent est lue puis acceptée seulement après lecture complète.
6. Chaque modification générée passe par une pull request relue par un autre développeur, avec des tests et des analyses automatiques au vert.
7. Le développeur reste responsable du code qu'il fusionne, qu'il l'ait écrit lui-même ou qu'il l'ait obtenu de l'agent après plusieurs essais.
8. Toute exposition accidentelle d'un secret déclenche aussitôt la procédure d'incident et l'information du référent chargé de la sécurité de l'équipe.
9. La consommation de crédits est suivie chaque mois, et un usage inhabituel est signalé sans attendre au responsable de l'équipe concernée.
10. Les questions, les cas limites et les idées d'amélioration remontent au référent désigné, qui met à jour cette charte à intervalle régulier.

Nom et prénom : Date : Signature :

Socle de configuration : l'exemple de la documentation officielle

Le fichier ci-dessous reprend l'exemple publié par Anthropic dans sa documentation sur les réglages gérés, consultée en 2026, et constitue un point de départ.

Il bloque la lecture de deux emplacements sensibles, désactive le mode de contournement et ignore les règles de permission posées par les développeurs.

Socle minimal : managed-settings.json

```
{
  "permissions": {
    "deny": [
      "Read(./.env)",
      "Read(./secrets/**)"
    ],
    "disableBypassPermissionsMode": "disable"
  },
  "allowManagedPermissionRulesOnly": true
}
```

Option : désactiver la télémétrie

Un second bloc, facultatif, coupe la télémétrie opérationnelle pour tous les développeurs que la politique atteint, au prix du tableau de bord d'analyse.

```
{
  "env": {
    "DISABLE_TELEMETRY": "1"
  }
}
```

Où déposer le fichier

Le fichier se dépose dans un répertoire système, dont l'emplacement varie selon le système d'exploitation de chaque poste de développement.

macOS	`/Library/Application Support/ClaudeCode/managed-settings.json`
Linux et WSL	`/etc/claude-code/managed-settings.json`
Windows	`C:\Program Files\ClaudeCode\managed-settings.json`

La documentation précise que Claude Code ne lit plus l'ancien chemin Windows situé sous ProgramData, ce qui peut expliquer une politique qui ne s'applique pas.

La commande `/status` affiche la source retenue sur la ligne Setting sources, et la commande `claude doctor` liste les entrées ignorées par Claude Code.

Votre bilan en trois lignes

Comptez les cases cochées dans les sept sections, puis lisez le niveau qui correspond à votre situation actuelle et aux décisions à prendre.

Les 19 points essentiels sont cochés
Le pilote peut démarrer sur une équipe restreinte, avec un suivi hebdomadaire des écarts et une revue formelle des points restés en recommandé.

Au moins un des 19 points essentiels reste ouvert
Le pilote attend : chaque point essentiel resté ouvert expose des données, du code ou des accès que l'entreprise ne maîtrise pas encore.

Moins de 21 cases cochées sur 42
Le cadrage reste à faire : reprenez la section sur le type de compte, puis organisez une revue commune avec la sécurité et le DPO.

Ce que cette checklist ne tranche pas

Six sujets dépassent le cadre d'une checklist, parce que la bonne réponse dépend de votre organisation, de vos contrats et de vos contraintes sectorielles.

Le choix du mode d'accès	Team, Enterprise, API ou fournisseur cloud : l'arbitrage dépend de vos contrats actuels, de votre hébergement et de votre politique de journalisation.
Une politique par groupe	La console claude.ai applique une seule politique à tous les développeurs, alors qu'une passerelle auto-hébergée la distribue par groupe d'identité.
Les sessions cloud et Cowork	Ces sessions ne lisent pas les politiques déposées sur le poste, et leur couverture demande un chemin de déploiement distinct.
Les secteurs réglementés	Banque, santé ou défense : l'analyse d'impact, le registre de traitement et la preuve de conformité exigent un dossier propre à chaque activité.
L'intégration à votre existant	La gestion des secrets, la supervision de sécurité et la chaîne d'intégration continue doivent s'articuler avec les nouveaux flux de l'agent.
La montée en compétence	Le niveau des équipes, leurs langages et leurs habitudes de revue déterminent le programme de formation et le rythme du déploiement.

Ces sujets se traitent mieux à partir de votre contexte réel que d'une liste générique, et ils conditionnent la solidité du déploiement.

Sources officielles consultées en 2026

- Anthropic, Claude Code : utilisation des données : https://code.claude.com/docs/en/data-usage
- Anthropic, Claude Code : réglages gérés : https://code.claude.com/docs/en/managed-settings
- Anthropic, tarifs et offres Claude : https://claude.com/pricing
- CNIL, fiches pratiques sur l'IA : https://cnil.fr/fr/les-fiches-pratiques-ia

Cette checklist constitue une base de travail et non un avis juridique, car la validation finale appartient à votre délégué à la protection des données.

Ethanolle, partenaire certifié Anthropic
Ethanolle accompagne les entreprises dans l'adoption de Claude, du cadrage des usages jusqu'à la formation des équipes de développement logiciel.
ethanolle.com