Bash – Lire un fichier d’IP ligne par ligne et appliquer une commande

Fiche A4 explicative (analyse + version corrigée + bonnes pratiques)

Ce que fait le script (objectif)

• Lire un fichier texte contenant une IP par ligne (ex. /home/flo/ip.txt)
• Pour chaque ligne non vide, exécuter une commande : **cscli decisions add --ip ...**
• Afficher (echo) la commande avant de l’exécuter (utile pour vérifier / auditer)

Script tel qu’on le voit (principe)

```
# Lire le fichier ligne par ligne
while IFS= read -r Ip
do
    # Vérifier si la ligne n'est pas vide
    if [[ -n "$ip" ]]; then
        echo "cscli decisions add --ip $ip"
        cscli decisions add --ip "$ip"
    fi
done < "/home/flo/ip.txt"
```

Analyse : points à corriger / risques

- Casse de variable incohérente : la lecture est dans « Ip » mais le test utilise « ip ». En Bash, Ip ≠ ip.
- Les lignes contenant des espaces, des retours chariot Windows (\r) ou des commentaires (#) peuvent provoquer des erreurs.
- Si le fichier n’a pas de saut de ligne final, le dernier élément peut être ignoré selon les cas (solution : condition de lecture robuste).
- Le script suppose que cscli existe et est dans le PATH ; en production, il faut vérifier (command -v).

Version corrigée (robuste, recommandée)

```
#!/bin/bash
set -euo pipefail
IFS=$'\n\t'

FILE="/home/flo/ip.txt"

# Vérifications de base
[ -f "$FILE" ] || { echo "Fichier introuvable: $FILE" >&2; exit 1; }
command -v cscli >/dev/null || { echo "cscli introuvable (PATH)" >&2; exit 1; }

# Lecture ligne par ligne (gère aussi le dernier ligne sans \n)
while IFS= read -r ip || [ -n "$ip" ]; do
    # Nettoyage : retire un éventuel \r (fichier Windows) et trim simple
    ip="${ip%$'\r'}"

    # Ignore lignes vides et commentaires
    [[ -z "$ip" || "$ip" =~ ^[[:space:]]*# ]] && continue

    echo "cscli decisions add --ip $ip"
    cscli decisions add --ip "$ip"
done < "$FILE"
```

Explication ligne par ligne (résumé)

Bloc	Rôle
while IFS= read -r ip	Lit une ligne sans interpréter les backslashes. IFS= empêche la suppression d’espace
|| [-n "$ip"]	Assure le traitement de la dernière ligne même sans saut de ligne final.
ip="${ip%$'\r'}"	Supprime un éventuel retour chariot Windows.
[[-z ...]] && continue	Ignore lignes vides et commentaires (#).
echo ... puis cscli ...	Affiche la commande (audit) puis l’exécute réellement.

Bonnes pratiques (Admin / DevOps)

- Toujours citer les variables : "$ip"
- Valider l’existence du fichier et de la commande avant d’agir
- Ajouter un mode « dry-run » si nécessaire (ex : variable DRY_RUN=1)
- Gérer les commentaires, lignes vides, et retours chariot Windows
- Tester la syntaxe : bash -n script.sh