

## 1. Préparation du projet

Récupérez le dépôt de démonstration et placez-vous dessus.

Créez deux branches pour vos tests :

`test-node` (pour npm)

`test-php` (pour Composer)

## 2. Mise en place du cache npm

Ouvrez le fichier de workflow dédié à Node.js et repérez l'étape de cache.

Sur la branche `test-node`, lancez votre première exécution sans modifier le lockfile.

Question 1 : le cache est-il restauré ou manqué ? Pourquoi ?

Apportez une modification mineure au lockfile (par exemple un commentaire ou un changement de version mineure), puis relancez l'exécution.

Question 2 : quel préfixe de `restore-keys` a permis la restauration ?

Modifiez significativement vos dépendances (ajout ou suppression d'un package), puis réexécutez le workflow.

Question 3 : comment se comporte la restauration quand la clé exacte n

Adaptez la liste des `restore-keys` pour tester une stratégie alternative (par exemple en basant vos préfixes sur le nom de la dépendance principale ou sur un fichier différent).

Question 4 : comparez le résultat avec la configuration initiale : que

## 3. Mise en place du cache Composer

Passez sur la branche `test-php` et ouvrez le workflow PHP.

Exécutez-le une première fois sans modification.

Question 5 : le dossier `vendor` est-il conservé ? En quelle circonstanc

Ajoutez une nouvelle dépendance dans votre configuration PHP, exécutez la mise à jour des paquets, puis relancez le workflow.

Question 6 : quel `restore-key` intervient pour récupérer le cache précé

Supprimez entièrement la section de restore-keys, testez à nouveau après une autre modification de dépendance.

Question 7 : quel impact sur la restauration ?

#### 4. Analyse et synthèse

Pour chaque exécution :

Relevez la mention « Cache restored from key » ou « Cache not found for ke

Notez si un restore-key a été utilisé.

Mesurez le temps gagné (installation VS absence de cache).