Mise à jour de node.js

Version de node.js EN

Pourquoi utiliser un gestionnaire de versions Node.js ?

Alors que l'installation directe de Node.js fonctionne parfaitement, vous rencontrerez rapidement une limitation majeure : vous ne pouvez avoir qu'une seule version de Node.js active globalement sur votre système à un instant T. Or, dans la réalité du développement, il est fréquent de devoir jongler entre différentes versions. Un projet ancien peut nécessiter une version LTS spécifique, tandis qu'un nouveau projet pourrait bénéficier des dernières fonctionnalités de la version “Current”. De même, vous pourriez vouloir tester la compatibilité de votre code avec une future version LTS.

C'est là qu'intervient NVM (Node Version Manager). NVM est un outil en ligne de commande qui vous permet d'installer, de gérer et de basculer facilement entre plusieurs versions de Node.js sur la même machine, sans interférences. Chaque version est isolée, avec ses propres modules globaux (si vous en installez). L'utilisation de NVM est considérée comme une bonne pratique, offrant flexibilité et évitant les conflits de versions entre projets.

Note importante : Le projet NVM original cible principalement les environnements macOS et Linux (ou WSL sous Windows). Pour les utilisateurs de Windows natif, un projet distinct mais très similaire, nvm-windows, offre des fonctionnalités équivalentes. Nous couvrirons l'installation pour les deux cas.

Installation de NVM

Pour macOS et Linux (via NVM original) :

L'installation se fait généralement via un script téléchargé avec `curl` ou `wget`. Ouvrez votre terminal et exécutez la commande d'installation fournie sur le dépôt GitHub officiel de NVM (github.com/nvm-sh/nvm). La commande ressemble généralement à ceci (vérifiez toujours la dernière version sur le site officiel) :

```
  curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh |
bash
  # Ou avec wget:
  wget -qO- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh
| bash
```

Ce script télécharge NVM dans un répertoire caché (`~/.nvm` par défaut) et tente de mettre à jour votre fichier de configuration de shell (`~/.bash_profile`, `~/.zshrc`, `~/.profile`, ou `~/.bashrc`) pour charger NVM automatiquement au démarrage d'une nouvelle session de terminal.

Après l'exécution du script, vous devrez soit **fermer et rouvrir votre terminal**, soit

exécuter la commande indiquée par le script (souvent `source ~/.bashrc` ou équivalent) pour charger NVM dans la session actuelle. Pour vérifier l'installation, tapez :

```
command -v nvm
```

Si NVM est correctement installé, cette commande devrait afficher `nvm`.

Pour Windows (via nvm-windows) :

Le projet `nvm-windows` propose un installateur dédié.

- Rendez-vous sur le dépôt GitHub de nvm-windows : github.com/coreybutler/nvm-windows.
- Allez dans la section “Releases” (Versions) et téléchargez le fichier `nvm-setup.zip` de la dernière version.
- Décompressez l'archive et exécutez le fichier `nvm-setup.exe`.

Suivez les instructions de l'assistant d'installation. Il vous demandera où installer NVM et où stocker les différentes versions de Node.js.

- Une fois l'installation terminée, ouvrez une nouvelle invite de commandes (cmd) ou PowerShell avec les droits d'administrateur (important pour la première utilisation et parfois pour changer de version).
    - Vérifiez l'installation en tapant :

```
nvm version
```

Cela devrait afficher la version de nvm-windows installée.

Note : `nvm-windows` et NVM original sont des projets distincts et peuvent avoir des syntaxes ou comportements légèrement différents sur certaines commandes avancées, mais les commandes de base sont largement similaires.

Commandes NVM essentielles

Une fois NVM (ou nvm-windows) installé, voici les commandes les plus utiles à connaître :

- Lister les versions disponibles :

```
nvm ls-remote
nvm ls-remote --lts # Pour voir seulement les versions LTS
```

Installer une version spécifique de Node.js :

```
  nvm install 24         # Installe la version 24.x.y
  nvm install node       # Installe la dernière version stable de Node.js
(Current)
  nvm install --lts      # Installe la dernière version LTS disponible
  nvm install lts/hydrogen # Installe la dernière version de la ligne LTS
nommée "Hydrogen" (Node 18)
  nvm install lts/*        # Installe la dernière version de la ligne LTS la
plus récente
```

Lister les versions installées localement :

```
  nvm ls
```

Cette commande affiche toutes les versions de Node.js que vous avez installées via NVM. Elle indique également la version actuellement active (`→`) et la version par défaut (`default →`). Utiliser une version spécifique (pour la session de terminal actuelle) :

```
  nvm use 18.17.1
  nvm use 16
  nvm use --lts
```

La commande `nvm use` change la version de `node` et `npm` uniquement pour la session de terminal en cours. Fermez et rouvrez le terminal, et vous reviendrez à la version par défaut (ou aucune si aucune n'est définie).

Définir une version par défaut (pour les nouvelles sessions) :

```
  nvm alias default 18.17.1
  nvm alias default lts/hydrogen
  nvm alias default node
```

Cette commande définit la version qui sera automatiquement utilisée chaque fois que vous ouvrirez une nouvelle fenêtre de terminal.

Voir la version actuelle :

```
  nvm current
```

Désinstaller une version :

```
  nvm uninstall 14.20.0
```

Exécuter une commande avec une version spécifique sans changer la session :

```
nvm run 16.15.0 --version   # Execute `node --version` avec Node 16.15.0
nvm exec 18 node my_script.js # Execute my_script.js avec Node 18
```