

# Comment installer et utiliser NVM (Node Version Manager) sur Ubuntu 22.04

NVM, également appelé « Node Version Manager », est utilisé pour installer et gérer plusieurs versions de Node.js sous Linux. Il fournit un utilitaire de ligne de commande qui aide les développeurs à installer plusieurs versions de Node et à basculer entre plusieurs versions selon les exigences de leur projet. NVM est très utile si vous travaillez sur plusieurs projets nécessitant différentes versions de Node.

Dans cet article, nous allons vous montrer comment installer et utiliser NVM pour gérer Node.js sur Ubuntu 22.04.

## Conditions préalables

- Un serveur exécutant Ubuntu 22.04.
- Un mot de passe root est configuré sur le serveur.

## Installer la NVM

Par défaut, NVM n'est pas inclus dans le référentiel par défaut d'Ubuntu. Vous devrez donc l'installer à partir du script.

Tout d'abord, installez CURL et Gnupg2 avec la commande suivante :

```
apt-get install curl gnupg2 -y
```

Ensuite, exécutez la commande suivante pour télécharger et exécuter le script d'installation NVM :

```
curl https://raw.githubusercontent.com/nvm-sh/nvm/master/install.sh | bash
```

La commande ci-dessus installera NVM et définira tous les paramètres d'environnement requis dans le

fichier .bashrc. => `nvm` Nodes), vous pouvez les supprimer du nœud système comme suit :

```
$ nvm use system
$ npm uninstall -g a_module
```

=> Close and reopen your terminal to start using nvm or run the following to use it now: export NVM\_DIR="\$HOME/.nvm"

```
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion
```

Maintenant, activez tous les paramètres à l'aide de la commande suivante :

```
source ~/.bashrc
```

Maintenant, vérifiez la version NVM à l'aide de la commande suivante :

```
nvm --version
```

Vous devriez voir le résultat suivant :

```
0.39.1
```

## Installer Node.js avec NVM

NVM est maintenant installé sur votre système. Vous pouvez désormais utiliser la ligne de commande NVM pour installer n'importe quelle version de Node.js sur votre système.

Par exemple, pour installer la dernière version de Node.js, exécutez la commande suivante :

```
nvm install node
```

Vous devriez voir le résultat suivant :

```
Downloading and installing node v18.9.1...
Downloading https://nodejs.org/dist/v18.9.1/node-v18.9.1-linux-x64.tar.xz...
##### 100.0%Computing
checksum with sha256sum
Checksums matched!
Now using node v18.9.1 (npm v8.19.1)
Creating default alias: default -> node (-> v18.9.1)
```

Pour vérifier la version installée de Node.js, exécutez la commande suivante :

```
node --version
```

Vous devriez voir le résultat suivant :

```
v18.9.1
```

Si vous souhaitez installer la dernière version stable de Node.js, exécutez la commande suivante :

```
nvm install node --lts
```

Vous devriez voir le résultat suivant :

```
v18.9.1 is already installed.  
Now using node v18.9.1 (npm v8.19.1)
```

Pour installer la version spécifique de Node.js (12.17.0), exécutez la commande suivante :

```
nvm install 12.17.0
```

Vous devriez voir le résultat suivant :

```
Downloading and installing node v12.17.0...  
Downloading https://nodejs.org/dist/v12.17.0/node-v12.17.0-linux-x64.tar.xz...  
#####  
##### 100.0%Computing checksum with sha256sum  
Checksums matched!  
Now using node v12.17.0 (npm v6.14.4)
```

Maintenant, vérifiez la version actuelle de Node.js à l'aide de la commande suivante :

```
node --version
```

Vous devriez voir le résultat suivant :

```
v12.17.0
```

## Utiliser NVM pour gérer les versions de Node.js

Pour répertorier toutes les versions de Node.js installées sur votre système, exécutez la commande suivante :

```
nvm ls
```

Vous devriez voir le résultat suivant :

```
-> v12.17.0  
v18.9.1  
système  
default -> node (-> v18.9.1)  
iojs -> N/A (default)  
unstable -> N/A (default)  
node -> stable (-> v18.9.1) (default)  
stable -> 18.9 (-> v18.9.1) (default)  
lts/* -> lts/gallium (-> N/A)  
lts/argon -> v4.9.1 (-> N/A)  
lts/boron -> v6.17.1 (-> N/A)  
lts/carbon -> v8.17.0 (-> N/A)  
lts/dubnium -> v10.24.1 (-> N/A)  
lts/erbium -> v12.22.12 (-> N/A)  
lts/fermium -> v14.20.1 (-> N/A)  
lts/gallium -> v16.17.1 (-> N/A)
```

Vous pouvez trouver toutes les versions de Node.js disponibles à l'aide de la commande suivante :

```
nvm ls-remote
```

Pour définir votre version par défaut de Node.js sur 12.17.0, exécutez la commande suivante :

```
nvm use 12.17.0
```

Vous devriez voir le résultat suivant :

```
Utilise maintenant le nœud v12.17.0 (npm v6.14.4)
```

Pour trouver la version par défaut de l'utilisateur actuel, exécutez la commande suivante :

```
nvm run default --version
```

Vous devriez voir le résultat suivant :

```
Nœud en cours d'exécution v16.9.0 (npm  
v7.21.1) v16.9.0
```

Vous pouvez également exécuter une application Node avec une version spécifique de Node.js à l'aide de la commande suivante :

```
nvm run v12.17.0 app.js
```

Pour supprimer une version spécifique de Node.js de votre système, exécutez la commande suivante :

## Conclusion

Dans ce guide, nous avons expliqué comment installer et utiliser NVM sur Ubuntu 22.04. Vous pouvez désormais utiliser la NVM pour gérer plusieurs versions de Node.js et basculer entre elles. J'espère que vous pourrez désormais exécuter votre application avec n'importe quelle version de Node.js.

---

