

Múltiples versiones de Node con NVM



Nicolás Avila [Follow](#)

May 2, 2018 · 4 min read

¡Nos encanta Node! Y cada vez que sale un nuevo release de una versión, no podemos esperar para probar todas las características geniales que trae. Si trabajamos con Node, de seguro tenemos una versión instalada en nuestro entorno de desarrollo. El problema ocurre cuando queremos trabajar con una versión específica en un proyecto o queremos probar una característica específica de un release: ¿qué estrategia podemos utilizar?. Ahí es cuando Node Version Manager o NVM sale al rescate. Esta pequeña utilidad nos permite mantener múltiples versiones de Node en nuestro sistema las cuales pueden ser cambiadas bajo demanda.

En este tutorial aprenderemos a instalar y usar NVM.



Breve introducción al versionamiento de Node

Node utiliza versionamiento semántico para cada uno de sus releases. Poseen un proceso muy estructurado para realizar el lanzamiento de una nueva versión.

El término LTS hace referencia a *Long Term Support*, es decir, una versión que actualmente se encuentra con soporte por un período de 18 meses. Luego de transcurrido ese tiempo, la versión pasará a modo mantención por 12 meses.

Al igual que Ubuntu, Node posee nombres en clave para algunas de sus versiones. Actualmente las siguientes se encuentran disponibles:

Argon: Node 4.x
Boron: Node 6.x
Carbon: Node 8.x

Instalando Node con NVM (Linux/macOS)

Este tutorial hace uso de la versión 0.33.11 de NVM. Para obtener la última versión disponible pueden visitar el repositorio del proyecto.

El primer paso es instalar NVM. Para ello debemos abrir un terminal y ejecutar lo siguiente:

```
curl -o-  
https://raw.githubusercontent.com/creationix/nvm/v0.33.11/install.sh |  
bash
```

cURL es una utilidad de línea de comando que nos permite transferir información a través de varios protocolos (HTTPS, HTTP, FTP, etc). Si cURL no está disponible en tu sistema puedes instalarla con `sudo [apt-get/yum] install curl` (Linux) o `brew install curl` (macOS).

Luego de instalar NVM deben cerrar y abrir nuevamente el terminal para hacer efectiva la instalación.

Usando NVM

Al abrir un nuevo terminal, ya tendremos disponible el comando `nvm`. si lo ejecutamos obtendremos algo similar a esto:

```
nico@neecohost ~$ nvm
Node Version Manager

Note: <version> refers to any version-like string nvm understands. This includes:
- full or partial version numbers, starting with an optional "v" (0.10, v0.1.2, v1)
- default (built-in) aliases: node, stable, unstable, iojs, system
- custom aliases you define with `nvm alias foo`

Any options that produce colored output should respect the `--no-colors` option.

Usage:
nvm --help                Show this message
nvm --version             Print out the installed version of nvm
nvm install [-s] <version> Download and install a <version>, [-s] from source. Uses .nvmrc if available
  --reinstall-packages-from=<version> When installing, reinstall packages installed in <node|iojs|node version number>
  --lts                     When installing, only select from LTS (long-term support) versions
  --lts=<LTS name>          When installing, only select from versions for a specific LTS line
  --skip-default-packages  When installing, skip the default-packages file if it exists
  --latest-npm             After installing, attempt to upgrade to the latest working npm on the given node version
nvm uninstall <version>   Uninstall a version
nvm uninstall --lts       Uninstall using automatic LTS (long-term support) alias `lts/*`, if available.
nvm uninstall --lts=<LTS name> Uninstall using automatic alias for provided LTS line, if available.
nvm use [--silent] <version> Modify PATH to use <version>. Uses .nvmrc if available
  --lts                   Uses automatic LTS (long-term support) alias `lts/*`, if available.
```

Escribe "nvm" en terminal favorito para ver todas las opciones disponibles

A continuación mostraré los comandos más utilizados. Pueden acceder a la lista completa de opciones disponibles escribiendo `nvm` en su terminal.

nvm list

Permite listar todas las versiones instaladas en tu sistema. En mi caso tengo instaladas las ver , 8.9.4 y 8.10.0.

```
nico@neecohost ~$ nvm list
v8.4.0
-> v8.9.1
v8.9.4
v8.10.0
default -> v8.9.1
node -> stable (-> v8.10.0) (default)
stable -> 8.10 (-> v8.10.0) (default)
iojs -> N/A (default)
lts/* -> lts/carbon (-> N/A)
lts/argon -> v4.9.1 (-> N/A)
lts/boron -> v6.14.1 (-> N/A)
lts/carbon -> v8.11.1 (-> N/A)
```

Versiones de Node instaladas en mi ambiente de desarrollo

nvm ls-remote --lts

Permite listar todas las versiones LTS disponibles para descargar desde los repositorios oficiales de Node. Si utilizan el comando sin el flag `--lts`,

NVM listará todas las versiones disponibles de Node.

```
nico@neecohost ~$ nvm ls-remote --lts
v4.2.0 (LTS: Argon)
v4.2.1 (LTS: Argon)
v4.2.2 (LTS: Argon)
v4.2.3 (LTS: Argon)
v4.2.4 (LTS: Argon)
v4.2.5 (LTS: Argon)
v4.2.6 (LTS: Argon)
v4.3.0 (LTS: Argon)
v4.3.1 (LTS: Argon)
v4.3.2 (LTS: Argon)
v4.4.0 (LTS: Argon)
v4.4.1 (LTS: Argon)
v4.4.2 (LTS: Argon)
v4.4.3 (LTS: Argon)
v4.4.4 (LTS: Argon)
v4.4.5 (LTS: Argon)
v4.4.6 (LTS: Argon)
v4.4.7 (LTS: Argon)
v4.5.0 (LTS: Argon)
v4.6.0 (LTS: Argon)
v4.6.1 (LTS: Argon)
v4.6.2 (LTS: Argon)
v4.7.0 (LTS: Argon)
v4.7.1 (LTS: Argon)
```

Versiones LTS listadas con el comando `nvm ls-remote --lts`

`nvm install <VERSIÓN>`

Permite instalar una versión en específico. Una vez ha finalizado la instalación, NVM automáticamente cambiará la versión a la última descargada.

```
nico@neecohost ~$ nvm install v8.11.1
Downloading and installing node v8.11.1...
Downloading https://nodejs.org/dist/v8.11.1/node-v8.11.1-darwin-x64.tar.xz...
##### 100.0%
Computing checksum with shasum -a 256
Checksums matched!
Now using node v8.11.1 (npm v5.6.0)
```

Instalando la versión 8.11.1 con nvm

nvm use <VERSIÓN>

Permite activar una versión específica de Node

```
nico@neecohost ~$ nvm use v8.9.1
Now using node v8.9.1 (npm v5.6.0)
```

Activando la versión 8.9.1 con nvm

nvm uninstall <VERSIÓN>

Permite desinstalar una versión específica de Node

```
nico@neecohost ~$ nvm uninstall v8.11.1  
Uninstalled node v8.11.1
```

Desinstalando la versión 8.11.1 con nvm

¿Que ocurre si ya tengo una versión de Node previamente instalada?

Es recomendable desinstalar previamente cualquier versión de Node antes de utilizar NVM.

¿Cómo sacarle partido a NVM en nuestros proyectos?

Puede ser un poco laborioso para algunos tener que estar cambiando constantemente de versión de Node cuando trabajamos en múltiples proyectos. Existe una solución que permite agilizar este proceso.

En la carpeta principal del proyecto se debe crear un archivo `.nvmrc` que contenga el número de versión que deseamos utilizar. Por ejemplo

8.9.1

Luego mediante la consola, nos dirigimos a la carpeta del proyecto y escribimos `nvm use` para activar la versión correspondiente.

¿Y si tengo Windows 🤖?

Lamentablemente NVM está solo disponible para Linux y macOS 😓... ¡Pero tranquil@s! 😊. Si tu sistema operativo es Windows aún puedes utilizar NVM. Existe una versión paralela (totalmente separada de la versión oficial) llamada `nvm-windows`. Los comandos básicos (como listar, instalar y usar) se encuentran disponibles.

[Nodejs](#)[Nvm](#)[Node For Beginners](#)

Discover Medium

Welcome to a place where words matter. On Medium, smart voices and original ideas take center stage - with no ads in sight. Watch

Make Medium yours

Follow all the topics you care about, and we'll deliver the best stories for you to your homepage and inbox. Explore

Become a member

Get unlimited access to the best stories on Medium — and support writers while you're at it. Just \$5/month. Upgrade

[About](#)[Help](#)[Legal](#)

