How to install NVM on Debian

NVM stands for Node Version Manager. With NVM, you can manage multiple Node.js instances of NodeJS and switch between them without uninstalling and reinstalling the Node.

NVM stands for Node Version Manager. With NVM, you can manage multiple Node.js instances of NodeJS and switch between them without uninstalling and reinstalling the Node.

This tutorial will guide you through installing and using NVM on a Debian 11 system.

Note: This guide was written for Debian 11 (Bullseye), but is also applicable to most other Debian-based distributions.

Prerequisites

1. A server running Debian 11
2. Non-root users have sudo privileges

How to install NVM on Debian

Step 1: Update the system

Before we start installing packages and making changes to the system, we should make sure that everything is up to date.

```
sudo apt update && sudo apt upgrade
```

It will take some time for this process to complete and then your system will be updated.

Step 2: Install NVM

Once the above command is executed, we can install NVM.

We will use the cURL package. To do this, we need to install it first.

```
sudo apt install curl -y
```

After that, we can download and run the installation file for NVM.

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

The above command downloads a short script from GitHub, runs it as root using bash, and installs NVM.

To apply the changes, we need to close and reopen our terminal or run the following command.

```
source ~/.profile
```

Again, this shouldn't take too long, and when it's done, we can verify that NVM is installed by running the command with the -v nvm argument. We cannot use the command with nvm because it is a user-installed script. It is not an actual application on the system.

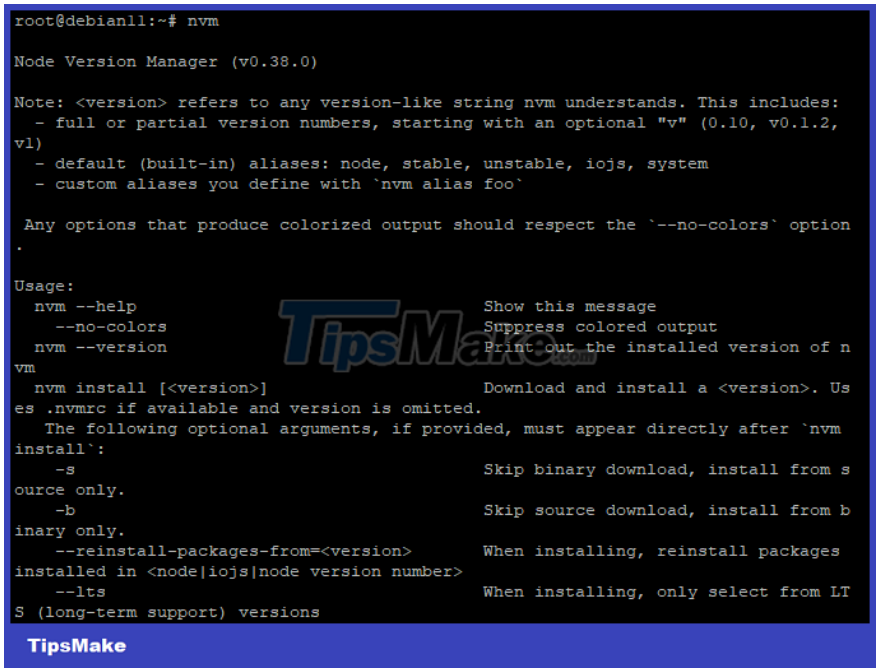
command -v nvm

If successful, you will see the following results. If not, repeat the installation steps above.

To check if the installation worked, you can run the following command, which will show you all the sub-commands available for NVM.

```
nvm
```

As you can see on the screenshot below, the installation was successful and sub-commands for NVM are available.



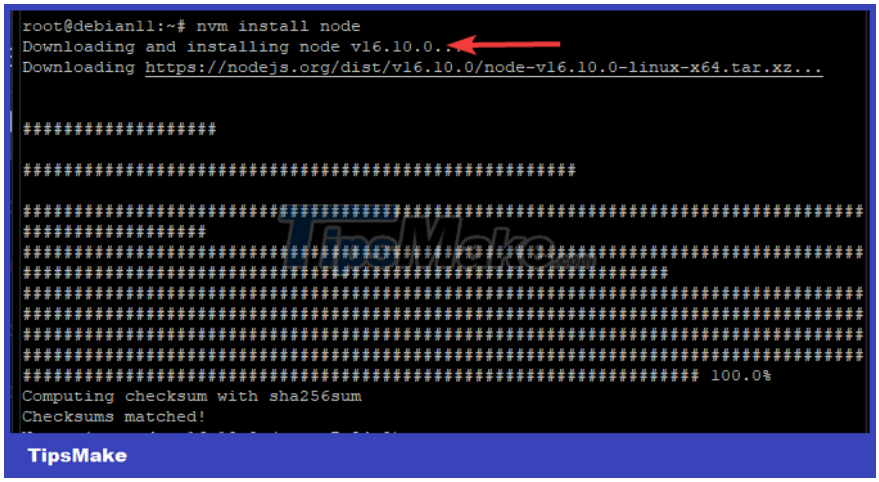
Step 3: Install Node.js with NVM

After installing NVM, installing Node.js is quite simple. With NVM, you can install multiple versions of Node.js under a single user account - you don't have to uninstall and reinstall Node.js or run into other hassles associated with multiple installations different versions of the same application.

To install the latest version of Node.js, run the following command.

```
nvm install node
```

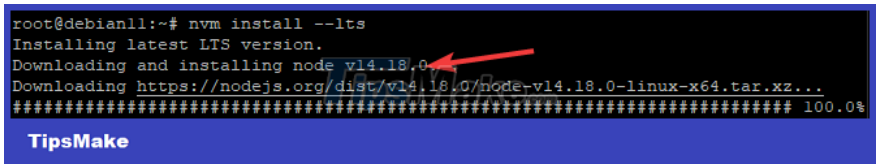
The command above uses NVM's standard installation mechanism to download and install the latest version of Node.js: v16.10.0 (at the time of writing).



To install the latest stable version of Node.js, you can use the --lts flag.

```
nvm install --lts
```

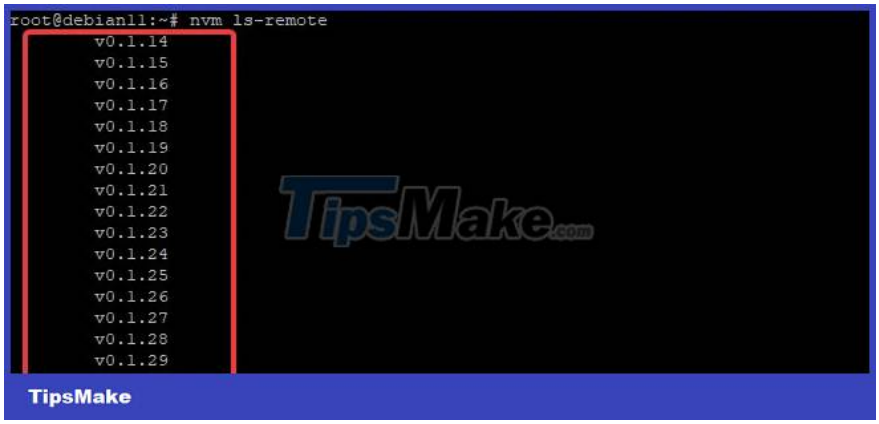
This command installs v14.18.0, the latest LTS version of Node.js at the time of writing.



To get a specific version of Node.js, you can use the nvm ls-remote command to get a list of all available versions, then select one from that list.

```
nvm ls-remote
```

The output of this command looks like the following. The screenshot below shows only a small portion of the versions in the list.



Once you've found the version you want, run the following command to install it.

```
nvm install
```

In there is the version of Node.js you want to install.

For example, if you want to install version 0.1.14, you need to run the following command.

```
nvm install 0.1.14
```

To get a complete list of all Node.js versions installed on your server, run the command below.

```
nvm ls
```

Sample output:

You can also switch to a different Node.js version for the current activeshell/user account.

```
nvm use
```

For example, to change Node.js version to v10.24.1, run the command below.

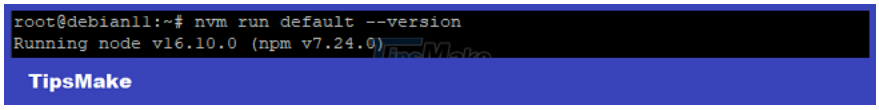
```
nvm use v10.24.1
```

Sample output:

To find the default version of Node.js that the current user account is using, run the following command.

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

Sample output:



Step 4: Test Node.js

So far, we've installed Node.js using NVM, creating a new Node.js instance, but how do we know if it's installed correctly?

In this step, we will create a simple Hello World project to test our Node.js installation with NVM.

To do this, create a file named hello.js in your home directory.

```
CD
```

```
sudo nano hello.js
```

Fill the file with the following content. Don't forget to save the file and exit when you're done by pressing CTRL + X, then Y then ENTER.

```
const http = require('http'); const hostname = 'localhost'; const port = 3000; c
```

Run your Node application using the command below.

```
nodejs hello.js
```

You should see the following output on the screen, letting you know that your Node.js application started correctly.

To test if your Node.js installation is working properly, open another terminal window on the same computer and try running the curl command below to output "Hello World!".

```
curl http://localhost:3000/
```

Your output will look something like this. If you don't get any errors, your Node.js installation is fine.

Now, remove your Node.js application using the rm command:

```
sudo rm -rf hello.js
```

Don't forget to exit the Node.js application by pressing CTRL + C. Otherwise, all further commands will be blocked. And the Node.js application running on your server may still be in a zombie state.

Congratulations! You have successfully installed and tested NVM and Node.js on your server.

You finished reading the article "**How to install NVM on Debian**" edited by the **TipsMake** team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.
