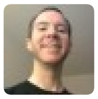
How to Install Node.js on Ubuntu 14.04

HOW-TO



By: Jacob Nicholson
Posted: May 10, 2016

Our experts take readers step-by-step through a variety of hosting and programming tasks in our popular series of "How-To" guides.

Node.js, a JavaScript-based platform, is an increasingly popular, lightweight, and efficient option for developers. It allows coders to use JavaScript on both the front-end and the back-end thanks to Google's awesome V8 engine.

With more than 100,000 packages, Node package manager, or npm, is a fantastic way to find open-source solutions for extending Node and npm's semantic versioning system, which allows for very tight control over what packages you are using.

5 Ways to Install Node.js on Ubuntu

There are several ways to do this, but **we recommend Option 1: Node Version Manager (nvm)**. Here is the full list of options:

Option 1 (Our pick!): Install the nvm script to manage multiple active Node.js versions

Option 2: Install the standard Debian/Ubuntu packages for Node and npm

Option 3: Install from Debian/Ubuntu packages created by the Node.js (associated) team

Option 4: Install Node.js manually from standard binary packages on the official website

Option 1 is our recommended method for everyone. Option 2 is incredibly simple, while Options 3, 4, and 5 have the advantage of keeping your Node and npm packages the most current. Before trying any of these install options, you'll want to remove the old Node package to avoid conflicts. And for any more information on Node.js, you can always check out the Github repository.

Before You Get Started: Remove Old Node Package to Avoid Conflicts

On Ubuntu, the Node.js package has a similar name to the older version, Node. The latter is an amateur packet radio program you can more than likely remove.

If you already have Node installed, you might want to remove it. Some Node.js tools might execute Node.js as Node instead of Node.js, causing conflicts.

You can look for and remove the Node package by executing these commands in a terminal. To access a terminal, navigate through the desktop menu:
Applications → Accessories → Terminal

Run this command and if it says install in the right column, Node is on your system:

```
$ dpkg --get-selections | grep node
ax25-node                                       install
node                                            install
```

If you found the old Node package installed, run this command to completely remove it:

```
sudo apt-get remove --purge node
```

Option 1: Install Node.js with Node Version Manager

First, make sure you have a C++ compiler. Open the terminal and install the build-essential and libssl-dev packages if needed. By default, Ubuntu does not come with these tools — but they can be installed in the command line.

Use apt-get to install the build-essential package:

```
sudo apt-get install build-essential checkinstall
```

Employ a similar process to get libssl-dev:

```
sudo apt-get install libssl-dev
```

You can install and update Node Version Manager, or nvm, by using cURL:

```
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.31.0/install.sh | bash
```

You will be asked to close and reopen the terminal. To verify that nvm has been successfully installed after you reopen the terminal, use:

```
command -v nvm
```

That command will output nvm if the installation worked.

To download, compile and install the latest version of Node:

```
nvm install 5.0
```

In any new shell, you'll need to tell nvm which version to use:

```
nvm use 5.0
```

To set a default Node.js version to be used in any new shell, use the *alias* default:

```
nvm alias default node
```

Not only does nvm allow you to run newer versions of Node.js and npm, you can install and migrate any desired versions you'd prefer. Go to the nvm GitHub repository for more information.

Option 2: Install Node.js with Ubuntu Package Manager

To install Node.js, type the following command in your terminal:

```
sudo apt-get install nodejs
```

Then install the Node package manager, npm:

```
sudo apt-get install npm
```

Create a symbolic link for node, as many Node.js tools use this name to execute.

```
sudo ln -s /usr/bin/nodejs /usr/bin/node
```

Now we should have both the Node and npm commands working:

```
$ node -v
v0.10.25
$ npm -v
1.3.10
```

Option 3: Install Node.js with Maintained Ubuntu Packages

Add the Node.js-maintained repositories to your Ubuntu package source list with this command:

```
curl -sL https://deb.nodesource.com/setup | sudo bash -
```

Then install Node.js with *apt-get*:

```
sudo apt-get install nodejs
```

Optionally we can create a symbolic link for node (for reasons mentioned earlier):

```
sudo ln -s /usr/bin/nodejs /usr/bin/node
```

Using this install option, we end up with newer versions of Node.js and npm:

```
$ node -v
v0.10.44
$ npm -v
2.15.0
```

Option 4: Install Node.js with Standard Binary Packages

Go to the official Node.js download page and download either the 32-bit or 64-bit Linux binary file, depending on your system type.

You can determine the CPU architecture of your server with these commands:

```
$ getconf LONG_BIT
64
$ uname -p
x86_64
```

You can download the file from the browser or from the console. The latter is shown below (Note: the specific Node.js version might be different for you):

```
wget https://nodejs.org/dist/v4.4.4/node-v4.4.4-linux-x64.tar.xz
```

To make sure you can unpack the file, install xz-utils:

```
sudo apt-get install xz-utils
```

Next, execute the following command to install the Node.js binary package in /usr/local/:

```
tar -C /usr/local --strip-components 1 -xJf node-v4.4.4-linux.x64.tar.xz
```

You should now have both Node.js and npm installed in /usr/local/bin. You can check this with:

```
ls -l /usr/local/bin/node
ls -l /usr/local/bin/npm
```

Final Words

Hopefully this will get you going with Node.js on Ubuntu. If you are new to developing applications with Node.js, the Nodeschool.io website has several interesting tutorials.