Community

≡ Menu

By: Nik van der Ploeg

⧉⁺ Subscribe

✔ How To Install Node.js with NVM (Node Version Manager) on a VPS

∧
♡
47

Posted July 19, 2013   👁 454k   NODE.JS

Status: Deprecated

This article is deprecated and no longer maintained.

Reason

Ubuntu 12.04 reached end of life (EOL) on April 28, 2017 and no longer receives security patches or updates.

See Instead

This article may still be useful as a reference, but may not follow best practices or work on this or other Ubuntu releases. We strongly recommend using a recent article written for the version of Ubuntu you are using.

- How To Install Node.js on Ubuntu 16.04

If you are currently operating a server running Ubuntu 12.04, we highly recommend upgrading or migrating to a supported version of Ubuntu:

- How to upgrade from Ubuntu 12.04 to Ubuntu 14.04.

- How to upgrade from Ubuntu 14.04 to Ubuntu 16.04

- How to migrate server data to a supported version

Introduction

on.

For those who haven't heard node.js is, it is the hot new cool kid on the block in web application development. It lets you write web apps that use Javascript on both the server and the client, so you don't need to know multiple programming languages to program your website. It's also really good at handling real-time concurrent web applications, which makes it a great choice for a lot of modern web apps.

The downside though is that all these cool new features are really, really *new*. As a result, getting up and running with node.js isn't as easy as, say, getting WordPress up and running on your web server.

This is the first in a series of how to install, code in, and use node. Joyent, the team behind node.js, has been improving node.js at a frantic pace, to the point where there are multiple releases of the software every month. For the most part, they've done a pretty good job of keeping things compatible; the things you write for one version of node will work just as well in the next. But nonetheless, sometimes a particular node app will only work with one version of node. And you will need to upgrade or downgrade your node.js install in order to use it.

This used to be a pain, but the node community has come together and created a great solution that lets you easily manage all your node installations and change node versions whenever you feel like it. It's called NVM, or the Node Version Manager.

Installing Node.js on a VPS

The install process couldn't be easier. Once you're logged into your VPS, run this command:

```
curl https://raw.githubusercontent.com/creationix/nvm/v0.11.1/install.sh | bas
```

You'll see some output fly by, and then nvm will be installed. You will see a line that says:

=> Close and reopen your terminal to start using NVM

It's not actually necessary to log out, we just need to make sure that the changes nvm made to your path are actually reflected, so just do:

```
source ~/.profile
```

Alternatively, run the command suggested in the output of the script. Now type:

```
nvm ls-remote
```

Should you see the error, `-bash: nvm: command not found` it may be because git is not installed.

Go ahead and install git and rerun the script:

```
apt-get install git
```

And you will be shown a list of all the available versions of node.js. You can always find out the latest stable release by heading to the node.js website, where it's printed in the center of the page.

To install version 0.10.13 (the latest as of this writing) type:

```
nvm install 0.10.13
```

If you type:

```
node --version
```

You will now see that node v0.10.13 is installed and active. If you had an older node app that only works with node v0.8.16, and wanted to downgrade, then you would input:

```
nvm install v0.8.16
```

to install and switch to v0.8.16.

When you're done and want to switch back to v0.10.13, you can do so with nvm's use command:

Nvm is great and makes switching between node versions easy and convenient. However, there's one caveat. If you type:

```
which node
```

you will see something interesting. Nvm installs node.js inside your user's home directory. This is fine for development, but if you want to actually host node applications, you don't want to install the latest new version of node via nvm and discover that you've inadvertently caused your production node app (which can be incompatible with the latest node.js) to stop working. It's best to install one copy of node globally so that other users can access it, and use nvm to switch between your development versions.

To do this, run the following command (entering your user's password at the prompt):

```
node);n=${n%/bin/node}; chmod -R 755 $n/bin/*; sudo cp -r $n/{bin,lib,share} /usr
```

◀   ▶

The above command is a bit complicated, but all it's doing is copying whatever version of node you have active via nvm into the /usr/local/ directory (where user installed global files should live on a linux VPS) and setting the permissions so that all users can access them.

> If you ever want to change the version of node that's installed system wide, just do another nvm use vXX.XX.XX to switch your user's node to the version you want, and then re-run the above command to copy it to the system directory.

To check that it works, become the root user and do another which command to make sure that node is now installed to /usr/local/bin:

```
sudo -s
which node
```

You should see:

```
/usr/local/bin/node
```

Sign up for our newsletter. Get the latest tutorials on SysAdmin and open source topics.   ✕

Enter your email address   Sign Up

By: Nik van der Ploeg   ♡ Upvote (47)   ⌐+ Subscribe

Spin up an SSD cloud server in under a minute.

Simple setup. Full root access.
Straightforward pricing.

DEPLOY SERVER

Related Tutorials

How To Use npm to Build and Publish Node.js Packages on a Linux Server

Geddy.JS: A No-Brainer MVC Node.js Framework

How To Use node.js, request and cheerio to Set Up Simple Web-Scraping

How To Install Express, a Node.js Framework, and Set Up Socket.io on a VPS

How To Use Winston to Log Node.js Applications

35 Comments

Leave a comment...

Log In to Comment

askmore *August 6, 2013*

0   Great article. Straightforward instructions.

---

erelsgl *September 29, 2013*

0   Great, thanks!

The only line that didn't work for me is "source ~/.bash_profile".

I (on Ubuntu 12.04) had to do "source ~/.bashrc" and "source ~/.profile" .

---

kamaln7   MOD   *September 29, 2013*

0   @erelsgl: Thanks, I've updated the article. It should be .profile.

---

erelsgl *October 2, 2013*

1   Another problem I just encountered is that, after reboot, the node version goes back to the previous one (see also this SO post: http://stackoverflow.com/questions/9170713/node-js-version-goes-back-to-0-4-form-0-6-on-reboot-nvm). To solve this, I had to add the line "nvm use xxx" to my .profile script.

---

jehoshua02 *October 6, 2013*

2   @erelsgl You could also do `nvm alias default XX.XX.XX`

---

yuvalkn1 *October 24, 2013*

0   From what I see, .profile starts nvm and that will set the $PATH correctly.

I have found that when using non-login shell (for example an SSH terminal), the .profile does not get sourced.
I found some info about how to deal with it:
1) http://superuser.com/questions/176404/bash-not-loading-profile-in-new-session-on-linux
2) http://superuser.com/questions/183870/difference-between-bashrc-and-bash-profile/183980#183980

[-s $HOME/.nvm/nvm.sh] && . $HOME/.nvm/nvm.sh # This loads NVM
from .profile to .bashrc

---

erelsgl *October 27, 2013*

0 In order to select the correct path anytime the server restarts, with or without console, I had
to put it in a "@reboot" record in the root crontab:

@reboot PATH=/root/.nvm/v0.10.18/bin:$PATH

---

me369802 *December 15, 2013*

0 Another way to solve this is to use nvm-global: https://github.com/xtuple/nvm

---

subfuzion *February 4, 2014*

0 Great article, nice and clear -- thanks!

---

scott460328 *March 6, 2014*

0 This tutorial didn't work on the LAMP droplet, it asked me to install git first (the tutorial says
you can do it later).
So I tried installing git and THAT didn't work

Here are the required steps that work for the LAMP stack image:
apt-get install git-core
(which fails)
apt-get update
(which works)
apt-get install git-core
(which now works)
curl https://raw.github.com/creationix/nvm/master/install.sh | sh
(which now works)
n=$(which node);n=${n%/bin/node}; chmod -R 755 $n/bin/*; sudo cp -r $n/{bin,lib,share}
/usr/local
(which fails)
cd ~
cp -r .nvm/v0.10.13/bin/node /usr/local
(which works)

I very much appreciate the effort Digital Ocean is making in these tutorials, but they are
                                                                                                              don't work
                                                                                                              as far from

nooitaf *March 13, 2014*

0 @scott it didn't say ubuntu or did he mention a droplet. If you would have used Arch or Fedora there wouldn't even be an apt-get, also explaining for every distro doesn't make a lot of sense and would just clutter the tutorial. Updating repos and installing apps should count before beginner level, as its the first thing you learn. Also Updating repos should be the first thing to do on any distro anyway as you can't expect DO to do daily updates of their droplets. But i agree that if you need git for the first command, it should say so in the beginning, but the 'copy' command of yours in not a good solution, as its explicit for one node version and not the one you are using right now. But maybe it should mention to move yourself to you home folder (cd ~) before executing.

madelineee3 *July 10, 2014*

0 NVM install script needs to be updated. Here is the newest curl, courtesy of the github page:

curl https://raw.githubusercontent.com/creationix/nvm/v0.10.0/install.sh | bash

mrhassell *July 11, 2014*

5 BE CAREFUL! The command listed above, is actually not well thought out and could lead to SUDO breaking (at the least) on many systems. Consider the following

```
chmod -R 755 $n/bin/*;
```

This recursively will change permissions on all executable files within /bin to 755, which is not a good thing in the case of SU or SUDO, which requires the sticky bit being set and will result in the following, unwanted behavior as the final part relocates the entire /bin to /usr/local/bin....

```
$ which sudo
/usr/local/bin/sudo
```

Perhaps a safer way to do this command, would be to remove the wildcard (*) and do something like this...

```
chmod 755 $n/bin/nodejs;  chmod 755 $n/bin/node; chmod 755 $n/bin/nvm; chmc
```

Even wild card on say /bin/n* would include   /usr/bin/nproc   used for limiting system processes

```
export NODE_PATH=/usr/local/bin/node
export NVM_DIR=/usr/local/bin/nvm
export PATH="$NODE_PATH/bin:$PATH;$NVM_DIR/bin:$PATH;"
```

This is not a complete example but using environment variables is a lot safer than changing system binary permissions and copying these without the correct permissions, setuid, stickybit etc... I appreciate the article and thanks for sharing! :-)

---

thyme.cypher *July 12, 2014*

0   curl https://raw.github.com/creationix/nvm/master/install.sh | sh

should be

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

---

kevin.dice1 *July 12, 2014*

0   This helped a lot. Thanks.

---

ljharb *July 19, 2014*

1   Hi - I maintain nvm. Please update your link to use the latest tagged release (per the readme on the github repo). Installing from `master` is dangerous, as you may install unstable code. Similarly, please pipe it to bash and not to sh (as is in the readme).

---

kamaln7 MOD   *July 19, 2014*

0   I've updated the command, thanks!

---

twendlan *July 26, 2014*

0   hmm does following the above mean you can only install new packages under root permissions? I previously had installed with npm but it was in my users home directory, which I understand is not ideal, `n=$(which node);n=${n%/bin/node}; chmod -R 755 $n/bin/*; sudo cp -r $n/{bin,lib,share} /usr/local` command to move it to the `/usr/local/bin` folder. Now I can no longer install packages, like ember-cli, without using sudo, which you are not supposed to do according to their documentation.

---

g $ node

`--version` gets me `bash: node: command not found`. Anyone got any thoughts on this?

⌃ [kamaln7](#) **MOD**   *February 8, 2015*

♡

0 Try adding the following line to `/home/youruser/.bashrc`:

        [-s $HOME/.nvm/nvm.sh] && source $HOME/.nvm/nvm.sh

You can then either run it manually to log out and log back in in order to have node added properly to your $PATH.

⌃ [lenswipe](#)   *February 8, 2015*

♡

0 [deleted]

⌃ [lenswipe](#)   *February 8, 2015*

♡

0 Yeah, I did that but when I open my shell I now get

    bash: [/home/<my username>/.nvm/nvm.sh: No such file or directory

Load More Comments

Community   Tutorials   Questions   Projects   Tags   Newsletter   RSS 🔊

Distros & One-Click Apps   Terms, Privacy, & Copyright   Security   Report a Bug   Write for DOnations   Shop

Sign Up