

Branch: **master** ▼

Find file

Copy path

**distributions** / README.md**SoundAxis** Add 19.04 to list of supported Ubuntu versions

019aad7 8 days ago

[33 contributors](#)

Raw

Blame

History



471 lines (318 sloc) 19.7 KB

# NodeSource Node.js Binary Distributions



ubuntu



redhat

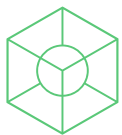


CentOS

fedora

PASSED

This repository contains documentation for using the **NodeSource Node.js** Binary Distributions via .rpm, .deb and Snap packages as well as their setup and support scripts.

**N|Solid Runtime**  
NODESOURCE

If you are looking for NodeSource's Enterprise-grade Node.js platform, **N|Solid**, please visit <https://downloads.nodesource.com/>

Please file an issue if you are experiencing a problem or would like to discuss something related to the distributions.

Pull requests are encouraged if you have changes you believe would improve the setup process or increase compatibility across Linux distributions.

## Table of Contents

- [Debian and Ubuntu based distributions](#) (deb)
  - [Installation instructions](#)
  - [Manual installation](#)
- [Enterprise Linux based distributions](#) (rpm)
  - [Installation instructions](#)
- [Snap packages](#)
  - [About](#)
  - [Installation instructions](#)
  - [Advanced usage](#)
- [Tests](#)
- [FAQ](#)
- [Requested Distributions](#)
- [License](#)

## Debian and Ubuntu based distributions

---

### Available architectures:

NodeSource will continue to maintain the following architectures and may add additional ones in the future.

- **i386** (32-bit)—**not available for Node.js 10 and later**
- **amd64** (64-bit)
- **armhf** (ARM 32-bit hard-float, ARMv7 and up: *arm-linux-gnueabihf*)
- **arm64** (ARM 64-bit, ARMv8 and up: *aarch64-linux-gnu*)

### Supported Ubuntu versions:

NodeSource will maintain Ubuntu distributions in active support by Canonical, including LTS and the intermediate releases.

- **Ubuntu 14.04 LTS** (Trusty Tahr) - **not available for Node.js 10 and later**
- **Ubuntu 16.04 LTS** (Xenial Xerus)
- **Ubuntu 18.04 LTS** (Bionic Beaver)
- **Ubuntu 18.10** (Cosmic Cuttlefish)
- **Ubuntu 19.04** (Disco Dingo)

### Supported Debian versions:

NodeSource will maintain support for stable, testing and unstable releases of Debian, due to the long release cycle a considerable number of users are running unstable.

- **Debian 8 / oldstable** (Jessie)

- **Debian 9 / stable** (Stretch)
- **Debian 10 / testing** (Buster)
- **Debian unstable** (Sid)

#### Supported Linux Mint versions:

- **Linux Mint 17 "Qiana"** (via Ubuntu 14.04 LTS) - **not available for Node.js 10 and later**
- **Linux Mint 17.1 "Rebecca"** (via Ubuntu 14.04 LTS) - **not available for Node.js 10 and later**
- **Linux Mint 17.2 "Rafaela"** (via Ubuntu 14.04 LTS) - **not available for Node.js 10 and later**
- **Linux Mint 18 "Sarah"** (via Ubuntu 16.04 LTS)
- **Linux Mint 18.1 "Serena"** (via Ubuntu 16.04 LTS)
- **Linux Mint 18.2 "Sonya"** (via Ubuntu 16.04 LTS)
- **Linux Mint 18.3 "Sylvia"** (via Ubuntu 16.04 LTS)
- **Linux Mint Debian Edition (LMDE) 2 "Betsy"** (via Debian 8)
- **Linux Mint 19 "Tara"** (via Ubuntu 18.04 LTS)

#### Supported Devuan versions:

- **Jessie** (via Debian 8)
- **Ascii** (via Debian 9)
- **Ceres** (via Debian unstable)

#### Supported elementary OS versions:

- **elementary OS Freya** (via Ubuntu 14.04 LTS) - **not available for Node.js 10 and later**
- **elementary OS Loki** (via Ubuntu 16.04 LTS)
- **elementary OS Juno** (via Ubuntu 18.04 LTS)

#### Supported Trisquel versions:

- **Trisquel 7 "Belenos"** (via Ubuntu 14.04 LTS) - **not available for Node.js 10 and later**
- **Trisquel 8 "Flidas"** (via Ubuntu 16.04 LTS)

#### Supported BOSS versions:

- **BOSS 5.0 "Anokha"** (via Debian 7) - **not available for Node.js 10 and later**
- **BOSS 6.0 "Anoop"** (via Debian 8)

#### Supported BunsenLabs versions:

- **Hydrogen** (via Debian 8)
- **Helium** (via Debian 9)

## Installation instructions

### Node.js v12.x:

```
# Using Ubuntu
curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -
sudo apt-get install -y nodejs

# Using Debian, as root
curl -sL https://deb.nodesource.com/setup_12.x | bash -
apt-get install -y nodejs
```

### Node.js v11.x:

```
# Using Ubuntu
curl -sL https://deb.nodesource.com/setup_11.x | sudo -E bash -
sudo apt-get install -y nodejs

# Using Debian, as root
curl -sL https://deb.nodesource.com/setup_11.x | bash -
apt-get install -y nodejs
```

### Node.js v10.x:

```
# Using Ubuntu
curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -
sudo apt-get install -y nodejs

# Using Debian, as root
curl -sL https://deb.nodesource.com/setup_10.x | bash -
apt-get install -y nodejs
```

### Node.js v8.x:

```
# Using Ubuntu
curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -
sudo apt-get install -y nodejs

# Using Debian, as root
curl -sL https://deb.nodesource.com/setup_8.x | bash -
apt-get install -y nodejs
```

### Node.js v6.x:

```
# Using Ubuntu
curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash -
sudo apt-get install -y nodejs

# Using Debian, as root
curl -sL https://deb.nodesource.com/setup_6.x | bash -
apt-get install -y nodejs
```

### **Optional:** install build tools

To compile and install native addons from npm you may also need to install build tools:

```
# use `sudo` on Ubuntu or run this as root on debian
apt-get install -y build-essential
```

## Manual installation

If you're not a fan of `curl <url> | bash -`, or are using an unsupported distribution, you can try a manual installation.

These instructions assume `sudo` is present, however some distributions do not include this command by default, particularly those focused on a minimal environment. In this case, you should install `sudo` or `su` to root to run the commands directly.

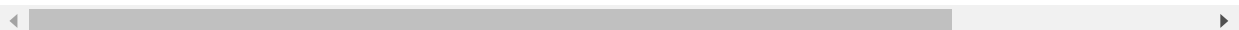
### 1. Remove the old PPA if it exists

This step is only required if you previously used Chris Lea's Node.js PPA.

```
# add-apt-repository may not be present on some Ubuntu releases:
# sudo apt-get install python-software-properties
sudo add-apt-repository -y -r ppa:chris-lea/node.js
sudo rm -f /etc/apt/sources.list.d/chris-lea-node_js-*.list
sudo rm -f /etc/apt/sources.list.d/chris-lea-node_js-*.list.save
```

### 2. Add the NodeSource package signing key

```
curl -sSL https://deb.nodesource.com/gpgkey/nodesource.gpg.key | sudo apt-
# wget can also be used:
# wget --quiet -O - https://deb.nodesource.com/gpgkey/nodesource.gpg.key |
```



### 3. Add the desired NodeSource repository

```
# Replace with the branch of Node.js or io.js you want to install: node_6.
VERSION=node_8.x
# The below command will set this correctly, but if lsb_release isn't avai
```

```
# - For Debian distributions: jessie, sid, etc...
# - For Ubuntu distributions: xenial, bionic, etc...
# - For Debian or Ubuntu derived distributions your best option is to use
DISTRO="$(lsb_release -s -c)"
echo "deb https://deb.nodesource.com/$VERSION $DISTRO main" | sudo tee /et
echo "deb-src https://deb.nodesource.com/$VERSION $DISTRO main" | sudo tee
```

#### 4. Update package lists and install Node.js

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

## Enterprise Linux based distributions

---

### Available architectures:

NodeSource will continue to maintain the following architectures and may add additional ones in the future.

- **i386** (32-bit)—not available for all distros and **not available for Node.js 10 and later**
- **x86\_64** (64-bit)

### Supported Red Hat® Enterprise Linux® versions:

- **RHEL 6** (64-bit)
- **RHEL 7** (64-bit)

### Supported CentOS versions:

- **CentOS 6** (64-bit)
- **CentOS 7** (64-bit)

### Supported CloudLinux versions:

- **CloudLinux 6** (32-bit for Node <= 10.x and 64-bit)

### Supported Fedora versions:

- **Fedora 27** (32-bit for Node <= 10.x and 64-bit)
- **Fedora 28** (64-bit)
- **Fedora 29** (64-bit) for Node >= 10.13.0

Equivalent versions of Korora Linux should also be supported.

## Installation instructions

*NOTE: If you are using RHEL 6 or CentOS 6, you might want to read about [running Node.js on older distros](#).*

Run as root on RHEL, CentOS, CloudLinux or Fedora:

### NodeJS 12.x

```
curl -sL https://rpm.nodesource.com/setup_12.x | bash -
```

### NodeJS 11.x

```
curl -sL https://rpm.nodesource.com/setup_11.x | bash -
```

### NodeJS 10.x

```
curl -sL https://rpm.nodesource.com/setup_10.x | bash -
```

### NodeJS 8.x

```
curl -sL https://rpm.nodesource.com/setup_8.x | bash -
```

### NodeJS 6.x

```
curl -sL https://rpm.nodesource.com/setup_6.x | bash -
```

**Optional:** install build tools

To compile and install native addons from npm you may also need to install build tools:

```
yum install gcc-c++ make  
# or: yum groupinstall 'Development Tools'
```

## Snap packages

---



## About

[Snaps](#) are containerized software packages designed to work across cloud, desktop, and IoT devices. They work natively on most popular Linux distributions and feature automatic transactional updates.

The NodeSource-managed Node.js snap contains the Node.js runtime, along the two most widely-used package managers, npm and Yarn. They are delivered from the [snapcraft store](#) and are automatically built and pushed for each supported Node.js release line. Generally you will have a new version of Node.js automatically running on your computer the same day it is released on [nodejs.org](#).

The Node.js snap can currently be installed on Arch Linux, Debian, Fedora, Linux Mint, Manjaro, OpenEmbedded/Yocto, OpenWrt, Solus, Ubuntu and many other distributions built on top these. NodeSource has not tested the Node.js snap on all of these distributions and feedback is welcome in this repository if you run into problems.

## Installation instructions

The `snap` command ships with Ubuntu, from version 16.04 and later. If you do not have it installed, follow the instructions on snapcraft to install [snapd](#).

Snaps are delivered via "channels", for Node.js, the channel names are the major-version number of Node.js. So select a supported Node.js version and install with:

```
sudo snap install node --classic --channel=8
```

Substituting `8` for the major version you want to install. Both LTS and Current versions of Node.js are available via snapcraft.

The `--classic` argument is required here as Node.js needs full access to your system in order to be useful, therefore it needs snap's "classic confinement". By default, snaps are much more restricted in their ability to access your disk and network and must request special access from you where they need it. Note that on some Linux distributions, the snap confinement mechanisms are not fully support so `--classic` may not be necessary or even supported.



Once installed, the `node`, `npm` and `yarn` commands are available for use and will remain updated for channel you selected.

## Switching channels

You can use the `refresh` command to switch to a new channel at any time:

```
sudo snap refresh node --channel=10
```

Once switched, `snapt` will update Node.js for the new channel you have selected.

## Bleeding-edge Node.js

Users feeling adventurous or interested in testing the latest code from the Node.js core developers can install from the "edge" channel. This has an element of risk: it is a direct pipeline from the upstream Node.js [git repository](#) to the snap store every day and previews the ongoing development work and may include breaking changes slated for the next major version of Node.js. This is only recommend for those users who are willing to participate in testing and bug reporting upstream:

```
sudo snap install node --classic --channel=edge
```

## Not recommended for production deployments

Due to their auto-updating nature, snaps are not necessarily appropriate for the deployment of your Node.js applications to production. NodeSource recommends a stable and integration-tested deployment pipeline for production applications such as the `.deb` or `.rpm` distributions outlined above. However, snaps are an excellent way to keep developer machines updated and allow for trivial and convenient switching between Node.js versions.

## Advanced usage

The `snap` man page, or Canonical's [advanced snap usage](#) tutorial contains details of advanced `snapt` functionality.

## Tests

---

To test an installation is working (and that the setup scripts are working!) use:

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

# FAQ

---

Q: How do I use this repo when behind a proxy?

A: Please take a look at [issue #9](#)

Q: How do I pin to specific versions of Node.js?

A: Please take a look at [issue #33](#)

Q: I upgraded to a new major version of Node.js using the scripts, but the old version is still being installed, what is going on?

A: You probably need to clear out your package manager's cache. Take a look at [issue #191](#)

Q: I'm trying to install Node.js on CentOS 5 / RHEL 5 and it is failing, why?

A: Due to the limitations of the compiler toolchain on EL 5 and its end of general support, we no longer support. See [issue #190](#)

Q: I'm seeing "Your distribution, identified as ".i686" or ".i386, is not currently supported, why?"

A: Node.js 4.x and newer require a 64bit os for rpms. See [issue #268](#)

Q: Why have certain versions of platforms/releases stopped receiving updates to Node.js?

A: Unfortunately, newer versions of V8 require a modern compiler toolchain. On some platforms, such as ARM wheezy, that toolchain is not available. See [issue #247](#)

Q: Why is my Node.js version newer than the one of the script I've run?

A: Your package manager is probably installing a newer Node.js version from a different source. See [issue #657](#)

Q: What is the current status of IPv6 support?

A: See [issue #170](#)

## Requested Distributions

We, unfortunately, do not have the resources necessary to support and test the plethora of Linux releases in the wild, so we rely on community members such as yourself to get support on your favorite distributions! This is a list of releases that have been requested by the community. If you are interested in contributing to this project, this would be a great place to start!

- OpenSUSE - [Issue #199](#)
- Scientific Linux - [Issue #251](#)
- LinuxMint Nadia - [Issue #269](#)
- TANGLU Bartholomea - [Issue #81](#)
- Korora - [Issue #130](#)
- FreePBX - [Issue #257](#)
- Deepin - [Issue #638](#)

## Authors and Contributors

<b>Chris Lea</b>	<a href="#">GitHub/chrislea</a>	<a href="#">Twitter/@chrislea</a>
<b>Rod Vagg</b>	<a href="#">GitHub/rvagg</a>	<a href="#">Twitter/@rvagg</a>
<b>William Blankenship</b>	<a href="#">GitHub/retrohacker</a>	<a href="#">Twitter/@retrohack3r</a>
<b>Harry Truong</b>	<a href="#">GitHub/harrytruong</a>	
<b>Matteo Brunati</b>	<a href="#">GitHub/mattbrun</a>	
<b>Brian White</b>	<a href="#">GitHub/mscdex</a>	
<b>Matt Lewandowsky</b>	<a href="#">GitHub/lewellyn</a>	
<b>Jan-Hendrik Peters</b>	<a href="#">GitHub/henr</a>	
<b>Andris Reinman</b>	<a href="#">GitHub/andris9</a>	
<b>Carvlsi</b>	<a href="#">GitHub/carvlsi</a>	
<b>Krasimir Trenchev</b>	<a href="#">GitHub/Ava7</a>	
<b>Phil Helm</b>	<a href="#">GitHub/phelma</a>	
<b>0xmohit</b>	<a href="#">GitHub/0xmohit</a>	

<b>jdaring</b>	<a href="#">GitHub/jdaring</a>	
<b>Prayag Verma</b>	<a href="#">GitHub/pa85</a>	
<b>Misha Brukman</b>	<a href="#">GitHub/mbrukman</a>	
<b>Simon Lydell</b>	<a href="#">GitHub/lydell</a>	
<b>Sebastian Blei</b>	<a href="#">GitHub/iamsebastian</a>	
<b>Jorge Maldonado Ventura</b>	<a href="#">NotABug/jorgesumle</a>	
<b>Mayank Metha</b>	<a href="#">GitHub/mayankmetha</a>	<a href="#">Twitter/@mayankmethad</a>

Contributions are welcomed from anyone wanting to improve this project!

## License

---

This material is Copyright (c) NodeSource and licensed under the MIT license. All rights not explicitly granted in the MIT license are reserved. See the included [LICENSE.md](#) file for more details.

*Supported with love by [Chris Lea](#), [Rod Vagg](#) and the [NodeSource](#) team*

*This project is not affiliated with Debian, Ubuntu, Red Hat, CentOS or Fedora.*

*Ubuntu is a registered trademark of Canonical Ltd.*

*Debian is a registered trademark owned by Software in the Public Interest, Inc.*

*Red Hat, CentOS and Fedora are trademarks of Red Hat, Inc.*

*CloudLinux is a trademark of Cloud Linux, Inc*