

OFFICIAL REPOSITORY

node (/r/ /node/) ☆

Last pushed: 15 hours ago

Repo Info (/ /node/)

### Short Description

Node.js is a JavaScript-based platform for server-side and networking applications.

### Full Description

## Supported tags and respective Dockerfile links

- 8.12.0-jessie , 8.12-jessie , 8-jessie , carbon-jessie , 8.12.0 , 8.12 , 8 , carbon (8/jessie/Dockerfile) (<https://github.com/nodejs/docker-node/blob/526c6e618300bdda0da4b3159df682cae83e14aa/8/jessie/Dockerfile>)
- 8.12.0-alpine , 8.12-alpine , 8-alpine , carbon-alpine (8/alpine/Dockerfile) (<https://github.com/nodejs/docker-node/blob/526c6e618300bdda0da4b3159df682cae83e14aa/8/alpine/Dockerfile>)
- 8.12.0-onbuild , 8.12-onbuild , 8-onbuild , carbon-onbuild (8/onbuild/Dockerfile) (<https://github.com/nodejs/docker-node/blob/526c6e618300bdda0da4b3159df682cae83e14aa/8/onbuild/Dockerfile>)
- 8.12.0-slim , 8.12-slim , 8-slim , carbon-slim (8/slim/Dockerfile) (<https://github.com/nodejs/docker-node/blob/526c6e618300bdda0da4b3159df682cae83e14aa/8/slim/Dockerfile>)
- 8.12.0-stretch , 8.12-stretch , 8-stretch , carbon-stretch (8/stretch/Dockerfile) (<https://github.com/nodejs/docker-node/blob/526c6e618300bdda0da4b3159df682cae83e14aa/8/stretch/Dockerfile>)
- 6.14.4-jessie , 6.14-jessie , 6-jessie , boron-jessie , 6.14.4 , 6.14 , 6 , boron (6/jessie/Dockerfile) (<https://github.com/nodejs/docker-node/blob/72dd945d29dee5afa73956ebc971bf3a472442f7/6/jessie/Dockerfile>)
- 6.14.4-alpine , 6.14-alpine , 6-alpine , boron-alpine (6/alpine/Dockerfile) (<https://github.com/nodejs/docker-node/blob/72dd945d29dee5afa73956ebc971bf3a472442f7/6/alpine/Dockerfile>)

- 6.14.4-onbuild , 6.14-onbuild , 6-onbuild , boron-onbuild  
(6/onbuild/Dockerfile) (<https://github.com/nodejs/docker-node/blob/72dd945d29dee5afa73956ebc971bf3a472442f7/6/onbuild/Dockerfile>)
- 6.14.4-slim , 6.14-slim , 6-slim , boron-slim (6/slim/Dockerfile)  
(<https://github.com/nodejs/docker-node/blob/72dd945d29dee5afa73956ebc971bf3a472442f7/6/slim/Dockerfile>)
- 6.14.4-stretch , 6.14-stretch , 6-stretch , boron-stretch  
(6/stretch/Dockerfile) (<https://github.com/nodejs/docker-node/blob/72dd945d29dee5afa73956ebc971bf3a472442f7/6/stretch/Dockerfile>)
- 11.2.0-alpine , 11.2-alpine , 11-alpine , alpine (11/alpine/Dockerfile)  
(<https://github.com/nodejs/docker-node/blob/241b2cfaca1ba225f312d439365d4410c94737fb/11/alpine/Dockerfile>)
- 11.2.0-slim , 11.2-slim , 11-slim , slim (11/slim/Dockerfile)  
(<https://github.com/nodejs/docker-node/blob/241b2cfaca1ba225f312d439365d4410c94737fb/11/slim/Dockerfile>)
- 11.2.0-stretch , 11.2-stretch , 11-stretch , stretch , 11.2.0 , 11.2 , 11 , latest (11/stretch/Dockerfile) (<https://github.com/nodejs/docker-node/blob/241b2cfaca1ba225f312d439365d4410c94737fb/11/stretch/Dockerfile>)
- 10.13.0-jessie , 10.13-jessie , 10-jessie , dubnium-jessie , 10.13.0 , 10.13 , 10 , dubnium (10/jessie/Dockerfile)  
(<https://github.com/nodejs/docker-node/blob/336fb229392876a5f0d893436aeccf8c80011eeb/10/jessie/Dockerfile>)
- 10.13.0-alpine , 10.13-alpine , 10-alpine , dubnium-alpine  
(10/alpine/Dockerfile) (<https://github.com/nodejs/docker-node/blob/336fb229392876a5f0d893436aeccf8c80011eeb/10/alpine/Dockerfile>)
- 10.13.0-slim , 10.13-slim , 10-slim , dubnium-slim (10/slim/Dockerfile)  
(<https://github.com/nodejs/docker-node/blob/336fb229392876a5f0d893436aeccf8c80011eeb/10/slim/Dockerfile>)
- 10.13.0-stretch , 10.13-stretch , 10-stretch , dubnium-stretch  
(10/stretch/Dockerfile) (<https://github.com/nodejs/docker-node/blob/336fb229392876a5f0d893436aeccf8c80011eeb/10/stretch/Dockerfile>)
- chakracore-8.11.1 , chakracore-8.11 , chakracore-8  
(chakracore/8/Dockerfile) (<https://github.com/nodejs/docker-node/blob/947280600648b70e067d35415d6812fd03127def/chakracore/8/Dockerfile>)
- chakracore-10.13.0 , chakracore-10.13 , chakracore-10 , chakracore  
(chakracore/10/Dockerfile) (<https://github.com/nodejs/docker-node/blob/69c8a5f448f46f9e34d7fb577eca79ba01f6864d/chakracore/10/Dockerfile>)

## Quick reference

- **Where to get help:**  
[the Docker Community Forums \(https://forums.docker.com/\)](https://forums.docker.com/), [the Docker Community Slack \(https://blog.docker.com/2016/11/introducing-docker-community-directory-docker-community-slack/\)](https://blog.docker.com/2016/11/introducing-docker-community-directory-docker-community-slack/), or [Stack Overflow \(https://stackoverflow.com/search?tab=newest&q=docker\)](https://stackoverflow.com/search?tab=newest&q=docker)
- **Where to file issues:**  
<https://github.com/nodejs/docker-node/issues> (<https://github.com/nodejs/docker-node/issues>)
- **Maintained by:**  
[The Node.js Docker Team \(https://github.com/nodejs/docker-node\)](https://github.com/nodejs/docker-node)
- **Supported architectures:** (more info (<https://github.com/docker-library/official-images#architectures-other-than-amd64>))  
[amd64 \(https://hub.docker.com/r/amd64/node/\)](https://hub.docker.com/r/amd64/node/), [arm32v6 \(https://hub.docker.com/r/arm32v6/node/\)](https://hub.docker.com/r/arm32v6/node/), [arm32v7 \(https://hub.docker.com/r/arm32v7/node/\)](https://hub.docker.com/r/arm32v7/node/), [arm64v8 \(https://hub.docker.com/r/arm64v8/node/\)](https://hub.docker.com/r/arm64v8/node/), [i386 \(https://hub.docker.com/r/i386/node/\)](https://hub.docker.com/r/i386/node/), [ppc64le \(https://hub.docker.com/r/ppc64le/node/\)](https://hub.docker.com/r/ppc64le/node/), [s390x \(https://hub.docker.com/r/s390x/node/\)](https://hub.docker.com/r/s390x/node/)
- **Published image artifact details:**  
[repo-info repo's repos/node/ directory \(https://github.com/docker-library/repo-info/blob/master/repos/node/\)](https://github.com/docker-library/repo-info/blob/master/repos/node/) ([history \(https://github.com/docker-library/repo-info/commits/master/repos/node/\)](https://github.com/docker-library/repo-info/commits/master/repos/node/))  
 (image metadata, transfer size, etc)
- **Image updates:**  
[official-images PRs with label library/node \(https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fnode\)](https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fnode)  
[official-images repo's library/node file \(https://github.com/docker-library/official-images/blob/master/library/node\)](https://github.com/docker-library/official-images/blob/master/library/node) ([history \(https://github.com/docker-library/official-images/commits/master/library/node\)](https://github.com/docker-library/official-images/commits/master/library/node))
- **Source of this description:**  
[docs repo's node/ directory \(https://github.com/docker-library/docs/tree/master/node\)](https://github.com/docker-library/docs/tree/master/node) ([history \(https://github.com/docker-library/docs/commits/master/node\)](https://github.com/docker-library/docs/commits/master/node))
- **Supported Docker versions:**  
[the latest release \(https://github.com/docker/docker-ce/releases/latest\)](https://github.com/docker/docker-ce/releases/latest) (down to 1.6 on a best-effort basis)

## What is Node.js?

Node.js is a software platform for scalable server-side and networking applications. Node.js applications are written in JavaScript and can be run within the Node.js runtime on Mac OS X, Windows, and Linux without changes.

Node.js applications are designed to maximize throughput and efficiency, using non-blocking I/O and asynchronous events. Node.js applications run single-threaded, although Node.js uses multiple threads for file and network events. Node.js is commonly used for real-time applications due to its asynchronous nature.

Node.js internally uses the Google V8 JavaScript engine to execute code; a large percentage of the basic modules are written in JavaScript. Node.js contains a built-in, asynchronous I/O library for file, socket, and HTTP communication. The HTTP and socket support allows Node.js to act as a web server without additional software such as Apache.

[wikipedia.org/wiki/Node.js](https://en.wikipedia.org/wiki/Node.js) (<https://en.wikipedia.org/wiki/Node.js>)



## How to use this image

See [How To Use This Image](https://github.com/nodejs/docker-node/blob/master/README.md#how-to-use-this-image) (<https://github.com/nodejs/docker-node/blob/master/README.md#how-to-use-this-image>) on GitHub for up-to-date documentation.

## Image Variants

The `node` images come in many flavors, each designed for a specific use case.

### `node:<version>`

This is the defacto image. If you are unsure about what your needs are, you probably want to use this one. It is designed to be used both as a throw away container (mount your source code and start the container to start your app), as well as the base to build other images off of.

This tag is based off of [buildpack-deps](https://hub.docker.com/_/buildpack-deps/) ([https://hub.docker.com/\\_/buildpack-deps/](https://hub.docker.com/_/buildpack-deps/)).

`buildpack-deps` is designed for the average user of Docker who has many images on their system. It, by design, has a large number of extremely common Debian packages. This reduces the number of packages that images that derive from it need to install, thus reducing the overall size of all images on your system.

Some of these tags may have names like `jessie` or `stretch` in them. These are the suite code names for releases of [Debian](https://wiki.debian.org/DebianReleases) (<https://wiki.debian.org/DebianReleases>) and indicate which release the image is based on.

## `node:<version>-alpine`

This image is based on the popular [Alpine Linux project](http://alpinelinux.org) (<http://alpinelinux.org>), available in the [alpine official image](https://hub.docker.com/_/alpine) ([https://hub.docker.com/\\_/alpine](https://hub.docker.com/_/alpine)). Alpine Linux is much smaller than most distribution base images (~5MB), and thus leads to much slimmer images in general.

This variant is highly recommended when final image size being as small as possible is desired. The main caveat to note is that it does use [musl libc](http://www.musl-libc.org) (<http://www.musl-libc.org>) instead of [glibc and friends](http://www.etalabs.net/compare_libcs.html) ([http://www.etalabs.net/compare\\_libcs.html](http://www.etalabs.net/compare_libcs.html)), so certain software might run into issues depending on the depth of their libc requirements. However, most software doesn't have an issue with this, so this variant is usually a very safe choice. See [this Hacker News comment thread](https://news.ycombinator.com/item?id=10782897) (<https://news.ycombinator.com/item?id=10782897>) for more discussion of the issues that might arise and some pro/con comparisons of using Alpine-based images.

To minimize image size, it's uncommon for additional related tools (such as `git` or `bash`) to be included in Alpine-based images. Using this image as a base, add the things you need in your own Dockerfile (see the [alpine image description](https://hub.docker.com/_/alpine/) ([https://hub.docker.com/\\_/alpine/](https://hub.docker.com/_/alpine/)) for examples of how to install packages if you are unfamiliar).

## `node:<version>-onbuild`

The `ONBUILD` image variants are deprecated, and their usage is discouraged. For more details, see [docker-library/official-images#2076](https://github.com/docker-library/official-images/issues/2076) (<https://github.com/docker-library/official-images/issues/2076>).

While the `onbuild` variant is really useful for "getting off the ground running" (zero to Dockerized in a short period of time), it's not recommended for long-term usage within a project due to the lack of control over *when* the `ONBUILD` triggers fire (see also [docker/docker#5714](https://github.com/docker/docker/issues/5714) (<https://github.com/docker/docker/issues/5714>), [docker/docker#8240](https://github.com/docker/docker/issues/8240) (<https://github.com/docker/docker/issues/8240>), [docker/docker#11917](https://github.com/docker/docker/issues/11917) (<https://github.com/docker/docker/issues/11917>)).

Once you've got a handle on how your project functions within Docker, you'll probably want to adjust your `Dockerfile` to inherit from a non-`onbuild` variant and copy the commands from the `onbuild` variant `Dockerfile` (moving the `ONBUILD` lines to the end and removing the `ONBUILD` keywords) into your own file so that you have tighter control over them and more transparency for yourself and others looking at your

Dockerfile as to what it does. This also makes it easier to add additional requirements as time goes on (such as installing more packages before performing the previously-ONBUILD steps).

## node:<version>-slim

This image does not contain the common packages contained in the default tag and only contains the minimal packages needed to run `node`. Unless you are working in an environment where *only* the `node` image will be deployed and you have space constraints, we highly recommend using the default image of this repository.

## License

View [license information \(https://github.com/nodejs/node/blob/master/LICENSE\)](https://github.com/nodejs/node/blob/master/LICENSE) for Node.js or [license information \(https://github.com/nodejs/docker-node/blob/master/LICENSE\)](https://github.com/nodejs/docker-node/blob/master/LICENSE) for the Node.js Docker project.

As with all Docker images, these likely also contain other software which may be under other licenses (such as Bash, etc from the base distribution, along with any direct or indirect dependencies of the primary software being contained).

Some additional license information which was able to be auto-detected might be found in [the `repo-info` repository's `node/` directory \(https://github.com/docker-library/repo-info/tree/master/repos/node\)](https://github.com/docker-library/repo-info/tree/master/repos/node).

As for any pre-built image usage, it is the image user's responsibility to ensure that any use of this image complies with any relevant licenses for all software contained within.

Docker Pull Command



```
docker pull node
```