

OFFICIAL REPOSITORY

jruby (/r/ _/ruby/) ☆

Last pushed: 5 days ago

[Repo Info \(/ _/ruby/\)](#)

Short Description

JRuby (<http://www.jruby.org>) is an implementation of Ruby (<http://www.ruby-lang.org>) on the JVM.

Full Description

Supported tags and respective Dockerfile links

- [latest](#) , [9](#) , [9.2](#) , [9.2.4](#) , [9.2-jre](#) , [9.2.4-jre](#) , [9.2.4.0](#) , [9.2.4.0-jre](#) ([9000/jre/Dockerfile](#)) (<https://github.com/cpuguy83/docker-jruby/blob/e80a6392db2062ef268fa7ef65c655b60219224b/9000/jre/Dockerfile>)
- [9-alpine](#) , [9.2-alpine](#) , [9.2.4-alpine](#) , [9.2-jre-alpine](#) , [9.2.4-jre-alpine](#) , [9.2.4.0-alpine](#) , [9.2.4.0-jre-alpine](#) ([9000/alpine-jre/Dockerfile](#)) (<https://github.com/cpuguy83/docker-jruby/blob/e80a6392db2062ef268fa7ef65c655b60219224b/9000/alpine-jre/Dockerfile>)
- [9-jdk](#) , [9.2-jdk](#) , [9.2.4-jdk](#) , [9.2.4.0-jdk](#) ([9000/jdk/Dockerfile](#)) (<https://github.com/cpuguy83/docker-jruby/blob/e80a6392db2062ef268fa7ef65c655b60219224b/9000/jdk/Dockerfile>)
- [9-jdk-alpine](#) , [9.2-jdk-alpine](#) , [9.2.4-jdk-alpine](#) , [9.2.4.0-jdk-alpine](#) ([9000/alpine-jdk/Dockerfile](#)) (<https://github.com/cpuguy83/docker-jruby/blob/e80a6392db2062ef268fa7ef65c655b60219224b/9000/alpine-jdk/Dockerfile>)
- [9-onbuild](#) , [9.2-onbuild](#) , [9.2.4-onbuild](#) , [9.2.4.0-onbuild](#) ([9000/onbuild/Dockerfile](#)) (<https://github.com/cpuguy83/docker-jruby/blob/e80a6392db2062ef268fa7ef65c655b60219224b/9000/onbuild/Dockerfile>)
- [9.1](#) , [9.1.17](#) , [9.1.17.0](#) , [9.1-jre](#) , [9.1.17-jre](#) , [9.1.17.0-jre](#) ([9000/jre/Dockerfile](#)) (<https://github.com/cpuguy83/docker-jruby/blob/8bc3fe27670a851953345182fe12f14f5e708383/9000/jre/Dockerfile>)
- [9.1-alpine](#) , [9.1.17-alpine](#) , [9.1.17.0-alpine](#) , [9.1-jre-alpine](#) , [9.1.17-jre-alpine](#) , [9.1.17.0-jre-alpine](#) ([9000/alpine-jre/Dockerfile](#))

<https://github.com/cpuguy83/docker->

[jruby/blob/8bc3fe27670a851953345182fe12f14f5e708383/9000/alpine-jre/Dockerfile](https://github.com/cpuguy83/docker-jruby/blob/8bc3fe27670a851953345182fe12f14f5e708383/9000/alpine-jre/Dockerfile))

- [9.1-jdk-alpine](https://github.com/cpuguy83/docker-jruby/blob/8bc3fe27670a851953345182fe12f14f5e708383/9000/alpine-jdk/Dockerfile) , [9.1.17-jdk-alpine](https://github.com/cpuguy83/docker-jruby/blob/8bc3fe27670a851953345182fe12f14f5e708383/9000/alpine-jdk/Dockerfile) , [9.1.17.0-jdk-alpine](https://github.com/cpuguy83/docker-jruby/blob/8bc3fe27670a851953345182fe12f14f5e708383/9000/alpine-jdk/Dockerfile) ([9000/alpine-jdk/Dockerfile](https://github.com/cpuguy83/docker-jruby/blob/8bc3fe27670a851953345182fe12f14f5e708383/9000/alpine-jdk/Dockerfile)) (<https://github.com/cpuguy83/docker-jruby/blob/8bc3fe27670a851953345182fe12f14f5e708383/9000/alpine-jdk/Dockerfile>)
- [9.1-jdk](https://github.com/cpuguy83/docker-jruby/blob/8bc3fe27670a851953345182fe12f14f5e708383/9000/jdk/Dockerfile) , [9.1.17-jdk](https://github.com/cpuguy83/docker-jruby/blob/8bc3fe27670a851953345182fe12f14f5e708383/9000/jdk/Dockerfile) , [9.1.17.0-jdk](https://github.com/cpuguy83/docker-jruby/blob/8bc3fe27670a851953345182fe12f14f5e708383/9000/jdk/Dockerfile) ([9000/jdk/Dockerfile](https://github.com/cpuguy83/docker-jruby/blob/8bc3fe27670a851953345182fe12f14f5e708383/9000/jdk/Dockerfile)) (<https://github.com/cpuguy83/docker-jruby/blob/8bc3fe27670a851953345182fe12f14f5e708383/9000/jdk/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/cpuguy83/docker-jruby/issues> (<https://github.com/cpuguy83/docker-jruby/issues>)

- **Maintained by:**

[the Docker Community](https://github.com/cpuguy83/docker-jruby) (<https://github.com/cpuguy83/docker-jruby>)

- **Supported architectures:** (more info (<https://github.com/docker-library/official-images#architectures-other-than-amd64>))

[amd64](https://hub.docker.com/r/amd64/jruby/) (<https://hub.docker.com/r/amd64/jruby/>), [arm32v5](https://hub.docker.com/r/arm32v5/jruby/) (<https://hub.docker.com/r/arm32v5/jruby/>), [arm64v8](https://hub.docker.com/r/arm64v8/jruby/) (<https://hub.docker.com/r/arm64v8/jruby/>), [i386](https://hub.docker.com/r/i386/jruby/) (<https://hub.docker.com/r/i386/jruby/>), [ppc64le](https://hub.docker.com/r/ppc64le/jruby/) (<https://hub.docker.com/r/ppc64le/jruby/>), [s390x](https://hub.docker.com/r/s390x/jruby/) (<https://hub.docker.com/r/s390x/jruby/>)

- **Published image artifact details:**

[repo-info](https://github.com/docker-library/repo-info/blob/master/repos/jruby/) [repo's](https://github.com/docker-library/repo-info/blob/master/repos/jruby/) [repos/](https://github.com/docker-library/repo-info/blob/master/repos/jruby/) [jruby/](https://github.com/docker-library/repo-info/blob/master/repos/jruby/) [directory](https://github.com/docker-library/repo-info/blob/master/repos/jruby/) (<https://github.com/docker-library/repo-info/blob/master/repos/jruby/>) ([history](https://github.com/docker-library/repo-info/commits/master/repos/jruby/) (<https://github.com/docker-library/repo-info/commits/master/repos/jruby/>))
(image metadata, transfer size, etc)

- **Image updates:**

[official-images](https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fjruby) [PRs](https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fjruby) with label [library/jruby](https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fjruby) (<https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fjruby>)
[official-images](https://github.com/docker-library/official-images/blob/master/library/jruby) [repo's](https://github.com/docker-library/official-images/blob/master/library/jruby) [library/jruby](https://github.com/docker-library/official-images/blob/master/library/jruby) [file](https://github.com/docker-library/official-images/blob/master/library/jruby) (<https://github.com/docker-library/official-images/blob/master/library/jruby>) ([history](https://github.com/docker-library/official-images/commits/master/library/jruby) (<https://github.com/docker-library/official-images/commits/master/library/jruby>))

- **Source of this description:**

[docs repo's jruby/ directory \(https://github.com/docker-library/docs/tree/master/jruby\)](https://github.com/docker-library/docs/tree/master/jruby)
([history \(https://github.com/docker-library/docs/commits/master/jruby\)](https://github.com/docker-library/docs/commits/master/jruby))

- **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 JRuby?

JRuby (<http://www.jruby.org> (<http://www.jruby.org>)) is an implementation of Ruby (<http://www.ruby-lang.org> (<http://www.ruby-lang.org>)) on the JVM.

Ruby is a dynamic, reflective, object-oriented, general-purpose, open-source programming language. According to its authors, Ruby was influenced by Perl, Smalltalk, Eiffel, Ada, and Lisp. It supports multiple programming paradigms, including functional, object-oriented, and imperative. It also has a dynamic type system and automatic memory management.

[wikipedia.org/wiki/Ruby_\(programming_language\)](https://en.wikipedia.org/wiki/Ruby_(programming_language))

([https://en.wikipedia.org/wiki/Ruby_\(programming_language\)](https://en.wikipedia.org/wiki/Ruby_(programming_language)))

JRuby leverages the robustness and speed of the JVM while providing the same Ruby that you already know and love. With JRuby you are able to take advantage of real native threads, enhanced garbage collection, and even import and use java libraries.



How to use this image

Create a `Dockerfile` in your Ruby app project

```
FROM jruby:9
```

```
# throw errors if Gemfile has been modified since Gemfile.lock  
RUN bundle config --global frozen 1
```

```
WORKDIR /usr/src/app
```

```
COPY Gemfile Gemfile.lock ./  
RUN bundle install
```

```
COPY . .
```

```
CMD ["/usr/src/app/your-daemon-or-script.rb"]
```

Put this file in the root of your app, next to the `Gemfile`.

You can then build and run the Ruby image:

```
$ docker build -t my-ruby-app .  
$ docker run -it --name my-running-script my-ruby-app
```

Generate a `Gemfile.lock`

The above example `Dockerfile` expects a `Gemfile.lock` in your app directory. This `docker run` will help you generate one. Run it in the root of your app, next to the `Gemfile`:

```
$ docker run --rm -v "$PWD":/usr/src/app -w /usr/src/app jruby:9 bundle install
```

Run a single Ruby script

For many simple, single file projects, you may find it inconvenient to write a complete `Dockerfile`. In such cases, you can run a Ruby script by using the Ruby Docker image directly:

```
$ docker run -it --rm --name my-running-script -v "$PWD":/usr/src/app jruby:9 bundle exec script.rb
```

Image Variants

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

`jruby:<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.

jruby:<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). 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), 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/) for examples of how to install packages if you are unfamiliar).

jruby:<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).

License

View [license information \(https://github.com/jruby/jruby/blob/master/COPYING\)](https://github.com/jruby/jruby/blob/master/COPYING) for the software contained in this image.

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 jruby/ directory \(https://github.com/docker-library/repo-info/tree/master/repos/jruby\)](https://github.com/docker-library/repo-info/tree/master/repos/jruby).

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 jruby
```