Extracted from:

Deploying with JRuby 9k

Deliver Scalable Web Apps Using the JVM

The Pragmatic Bookshelf

Raleigh, North Carolina

The Pragmatic Programmers

Deploying with JRuby 9k

Deliver Scalable Web Apps Using the JVM

Joe Kutner

Edited by Brian P. Hogan

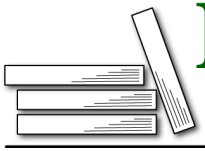
Deploying with JRuby 9k

Deliver Scalable Web Apps Using the JVM

Joe Kutner

The Pragmatic Bookshelf

Raleigh, North Carolina



Pragmatic Bookshelf

Our Pragmatic books, screencasts, and audio books can help you and your team create better software and have more fun. Visit us at *https://pragprog.com*.

The team that produced this book includes:

Brian P. Hogan (editor)
Potomac Indexing, LLC (index)
Linda Recktenwald (copyedit)
Gilson Graphics (layout)
Janet Furlow (producer)

For sales, volume licensing, and support, please contact *support@pragprog.com*.

For international rights, please contact *rights@pragprog.com*.

Printed in the United States of America.
ISBN-13: 978-1-68050-169-8
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—July 2016

Installing Docker

Deployment is the process of taking code or binaries from one environment and moving them to a another environment where you execute them. In the case of a web app case, you'll move code from your development machine to a production server. You've already configured a development environment, but you still need to create a production environment you can use as the target of your deployments. For this, you'll use Docker,[1] which reduces the process of provisioning a production environment to just a few steps.

Docker is a Linux-based containerization platform. It runs processes in isolated environments without the need for a complete visualization layer for each process. You can use Docker to run multiple isolated processes on the same host without excessive overhead. You'll use Docker primarily as a development tool, allowing you to run a simulated production environment on a local machine. There is one catch, though. If your local machine isn't running Linux, you'll need a virtual machine to run Docker. Fortunately, Docker distributes a lightweight headless virtual machine, called Docker Machine, that provides many advantages over traditional virtualization.

Docker Machine runs on VirtualBox,[2] an open source virtualization platform. If you're not running Linux, you'll need to install both of these. If you are running Linux, you'll need to install only Docker.

Installing Docker on Mac OS X or Windows

You can install both Docker and Docker Machine on Mac and Windows using the Docker Toolbox native installer. Open a browser and navigate to the Docker Toolbox website.[3] Select the Installer for Mac and download it. Once the package is downloaded, open it to run the wizard. Follow the steps in the wizard to complete the installation as described on the Docker website.[4]

Now open a terminal. If you're on Windows, you'll need to double-click the Docker CLI shortcut on your Desktop to open a Docker terminal instead of a standard terminal. If the system displays a User Account Control prompt to allow VirtualBox to make changes to your computer, choose Yes. From the terminal, run these commands:

```
$ docker-machine -v
```

1. http://docker.com/
2. https://www.virtualbox.org
3. https://www.docker.com/toolbox
4. http://docs.docker.com/mac/step_one/

```
docker-machine version 0.4.0 (9d0dc7a)
```

```
$ docker-machine ls
NAME      ACTIVE   DRIVER       STATE     URL
default   *        virtualbox   Running   tcp://192.168.99.100:2376
```

This shows that a single Docker Machine, named default, is running.

On Mac only, you must set a few environment variables so the Docker client can communicate with the Docker Machine. Run this command:

```
$ eval "$(docker-machine env default)"
```

On Windows, those environment variables are set for you when you run the Desktop app. On Mac, if you don't want to run that command every time you open a new terminal, then run this command to add the environment variables to your profile:

```
$ docker-machine env default >> ~/.profile
```

Now you're ready to use the Docker client. You can move on to *Getting Started with Docker*, on page ?.

Installing Docker on Linux

Docker runs natively on Ubuntu, but it requires a 64-bit architecture and a 3.10 kernel version or later. You can check your architecture by running this command:

```
$ uname -p
x86_64
```

And you can check your kernel version with this command:

```
$ uname -r
3.13.0-57-generic
```

If the output of either `uname` command doesn't match the requirements, then you'll need to run Docker on a virtualization layer by installing Docker Machine.[5] Otherwise, you can install Docker natively. To begin the native installation, update your package manager by running this command:

```
$ sudo apt-get update
```

Then install the generic Linux kernel image. This kernel has the advanced multi-layered unification filesystem (AUFS) built in, which is required to run Docker.

```
$ sudo apt-get install linux-image-generic-lts-trusty
```

5. http://docs.docker.com/machine/install-machine/

Now reboot your machine:

```
$ sudo reboot
```

After your computer has restarted, you can install Docker with this command:

```
$ curl -sSL https://get.docker.com/ | sh
```

Now you're ready to use the Docker CLI.