



# Creación de una imagen Docker con Ubuntu y Java

por Ivan K · 11 de agosto · Zona de nubes

Implementar y escalar aplicaciones ricas en datos en minutos y con facilidad . Mesosphere DC / OS incluye todo lo que necesita para ejecutar de forma elástica aplicaciones y servicios de datos en contenedores en producción.

Este breve artículo muestra cómo crear una imagen Docker con Ubuntu 15.04 y que tiene Java 7 u 8 instalado. Funcionará como una base para algunos futuros artículos que he planeado escribir.

Sí, soy consciente de que hay varias imágenes de Docker con Java 8 en Docker Hub .

## Requisitos previos

Al escribir este artículo, he utilizado Docker 1.7.1 y Docker-Machine 0.3.0. Aquí encontrará instrucciones sobre cómo instalarlas .

## Archivo Docker

La documentación de Docker recomienda que los archivos de Docker se ubiquen en un directorio dedicado que sólo contiene los archivos necesarios para la creación de la imagen de Docker.

Por lo tanto, he creado un directorio llamado "krizsan-ubuntu1504java8" para la imagen Docker con Java 8 y otro directorio llamado "krizsan-ubuntu1504java7" para la imagen con Java 7.

Cada uno de estos directorios debe contener un archivo llamado "Dockerfile" : Sin extensión de archivo!).

El archivo Docker para Java 8 en Ubuntu 15.04 tiene el siguiente contenido:

```
1  # Ubuntu 15.04 con Java 8 instalado .
2  # Construir imagen con : docker build -t krizsan / ubuntu1504java8 : v1 .
3
4  DESDE ubuntu : 15.04
5  MAINTAINER Ivan Krizsan , https : //github.com/krizsan
6  RUN apt - get update && \
7      Apt - get upgrade - y && \
8      Apt - get install - y software - propiedades - common && \
9      Add - apt - repositorio ppa : webupd8team / java - y && \
10     Apt - get update && \
11     Echo oracle - java7 - instalador compartido / aceptado - oracle - licencia - v1 - 1
12     Apt - get install - y oracle - java8 - instalador && \
```

```

12 apt - get install -y oracle-java8-installer && \
13 apt - conseguir limpia

```

La versión corta es que este archivo Docker creará una imagen de Docker basada en la imagen de Ubuntu: 15.04 Docker en la que está instalada Java 8 de Oracle.

Versión larga, fila por fila:

- FROM ubuntu: 15.04  
Estados que Docker imagen de esta imagen Docker se basa en. Compare con la clase padre en la programación orientada a objetos.
- MAINTAINER Ivan Krizsan, <https://github.com/krizsan>  
Nos dice quién está manteniendo esta imagen de Docker.
- El comando Docker RUN ejecuta comandos de shell para instalar programas, configurar la imagen de Docker, etc. A continuación se detallan los propósitos de cada comando de shell.
- Apt-get update  
Actualiza la lista de paquetes disponibles y sus versiones.
- Apt-get upgrade  
Instala las versiones más nuevas de los paquetes actualmente instalados.
- Apt-get install -y software-properties-common  
Instala el software para permitir fuentes independientes de software del proveedor, en particular el comando "add-apt-repository" que se utiliza en la siguiente línea.
- Add-apt-repository ppa:webupd8team / java -y  
Agrega el repositorio PPA que contiene Oracle Java 8, suponiendo que sí a todas las consultas.
- Apt-get update  
Una vez más, actualiza la lista de paquetes disponibles y sus versiones. La razón para ejecutar una actualización de nuevo es que se agregó un nuevo repositorio y queremos actualizar la lista de paquetes disponibles y sus versiones, incluidas las del nuevo repositorio.
- Echo oracle-java7-installer compartido / aceptado-oracle-license-v1-1 select true | / Usr / bin / debconf-set-selections  
El instalador de Java requiere que acepte una licencia antes de que comience la instalación. Esta línea automatiza la aceptación de la licencia.
- Apt-get install -y oracle-java8-installer  
Finalmente Java 8 puede ser instalado! El indicador -y ejecuta la instalación en un modo no interactivo, suponiendo que todas las preguntas deben responderse "sí".
- Apt-get clean  
Limpia el repositorio local de los archivos de paquetes recuperados.

La razón de tener un único comando RUN de Docker y encadenar los comandos de shell con && es que Docker creará una capa adicional en la imagen para cada comando RUN. Cuantos menos comandos RUN contenga un archivo Docker, menor será la imagen resultante.

El archivo Docker que creará una imagen con Java 7 en Ubuntu 15.04 se verá así:

```

1 # Ubuntu 15.04 con Java 7 instalado .

```

```

1
2 # Construir imagen con : docker build -t krizsan / ubuntu1504java7 : v1 .
3
4 DESDE ubuntu : 15.04
5 MAINTAINER Ivan Krizsan , https : //github.com/krizsan
6 RUN apt - get update && \
7     Apt - get upgrade - y && \
8     Apt - get install - y software - propiedades - common && \
9     Add - apt - repositorio ppa : webupd8team / java - y && \
10    Apt - get update && \
11    Echo oracle - java7 - instalador compartido / aceptado - oracle - licencia - v1 - 1
12    Apt - get install - y oracle - java7 - instalador && \
13    apt - conseguir limpia

```

La única diferencia con respecto al archivo Java 8 Docker, aparte de los comentarios, es la línea de segundo a último en la que se utiliza el instalador de Java 7.

Construya la imagen de Java 8 Docker siguiendo estos pasos:

- Abra una ventana de terminal.
- Si no está utilizando un sistema operativo Linux, cree una Docker-Machine (si aún no la tiene):  
docker-machine create dev
- Si no está utilizando un sistema operativo Linux, inicie el Docker-Machine:  
dev
- Si no está usando un sistema operativo Linux, ssh en el Docker-Machine:  
docker-machine ssh dev
- Vaya al directorio que contiene el archivo Docker que describe la imagen que desea construir.  
En este ejemplo, construiré la imagen de Java 8.
- Construir la imagen Docker:  
docker build -t krizsan / ubuntu1504java8: v1.  
¡Añote el período al final de la línea!  
Los usuarios de Linux tendrán que agregar el comando sudo antes del comando docker:  
sudo docker build -t krizsan / ubuntu1504java8: v1.  
Verá bastante salida de consola que debería terminar con algo como esto:

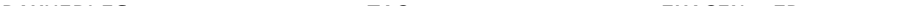
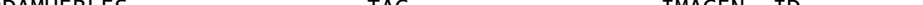
```

1 ...
2 Update - alternatives : use / usr / lib / jvm / java - 8 - oracle / bin / xjc para prop
3 Oracle JDK 8 instalado
4 ...
5 Invocar - rc . D : política - rc . D denegó la ejecución del inicio .
6 Ajuste hasta xfonts - codificaciones ( 1 : 1,0 . 4 - 2 ) ...
7 Configuración de xfonts - utils ( 1 : 7.7 + 2 ) ...
8 Configuración de gsfonts - x11 ( 0.22 ) ...
9 Procesamiento desencadena por libc - bin ( 2,21 - 0U buntu4 ) ...
10 Procesamiento desencadena por systemd ( 219 - 7U buntu6 ) ...

```

- Inicie un contenedor Docker usando la imagen Docker recién creada:  
docker run -it krizsan / ubuntu1504java8: v1 bash  
Una vez más, los usuarios de Linux necesitan insertar el comando sudo antes del comando docker.
- Compruebe la versión Java instalada:  
java -version  
En mi caso veo que la versión Java 1.8.0\_51 está instalada.
- Salga del contenedor Docker:  
salga

- Lista de las imágenes de Docker disponibles localmente: imágenes de docker

	GUARDAMUEBLES	TAG	IMAGEN	ID	CREADO
1					
	Krizzsán / ubuntu1504java8	v1	3f 1e02768	A70	16 segundos ha
2					

Esto concluye este artículo y ahora estoy listo para crear más imágenes de Docker que contengan software que se ejecuta en Java.

Descubre nuevas tecnologías que simplifican la ejecución de contenedores y servicios de datos en producción con este eBook gratuito de O'Reilly . Cortesía de Mesosphere .



**1/2 Day**  
**Getting to Know**  
**Google Compute**  
**Engine**  
 and How to Use It  
 STEVEN ADAMS, JR.

**Libre DZone Refcard**  
**Cómo conocer Google Compute Engine y cómo usarlo**

Publicado en DZone con permiso de Ivan K. Ver el artículo original aquí. 

Las opiniones expresadas por los contribuyentes de DZone son propias.

# Obtenga lo mejor de Cloud en su bandeja de entrada.

Manténgase actualizado con el boletín semanal de Cloud de DZone. VER UN EJEMPLO

SUSCRIBIR

## Cloud Partner Resources

Networking for Docker Containers: 3 Part Series

Mesosphere



Free report: How Mesos is enabling the adoption of containers and big data services

Mesosphere



Three key metrics for monitoring AWS SNS performance and usage

Site 24x7



Whitepaper: Site24x7: Powerful, Agile, Cost-Effective IT Management from the Cloud

Site 24x7



## Configure Bitbucket Webhook to Trigger Jenkins Builds on AWS EC2

by Ajitesh Kumar MVB · Aug 04, 17 · Cloud Zone

Site24x7 - Full stack It Infrastructure Monitoring from the cloud. Sign up for free trial.

This article represents the steps required to configure BitBucket Webhooks to trigger Jenkins Builds on AWS EC2 based on code committed in the repository. This essentially means that a code commit in the BitBucket code repository would trigger a build on the Jenkins server running on a AWS EC2 machine. This forms the starting point of a continuous delivery pipeline. The Jenkins build, when triggered as a result of code push, could perform tasks such as some of the following:

- Run the build.
- Run tests.
- Publish build artifacts in artifactory.
- Deploy the build artifacts in different environments including QA, UAT, and production.

Please feel free to comment/suggest if I missed one or more important points.

Following are the steps which need to be taken to configure the push-based trigger of a Jenkins build:

- **Configure Inbound Rule on Jenkins Server on AWS EC2:** One needs to add the inbound

Configure inbound rule on the Jenkins box on AWS EC2 to follow this IP: 104.192.143.192/28. This is required to allow requests coming from BitBucket to reach the Jenkins server on AWS EC2. This is the BitBucket cloud IP outbound address (for hooks like POST), while the inbound IP address for BitBucket is 104.192.143.1. The IP (104.192.143.1) can be figured out by pinging bitbucket.org. The outbound IP address can be found on this page.

- **Configure Webhook in BitBucket:** One needs to add the following URL while creating the WebHook in BitBucket: `http://jenkins_server_uri/bitbucket-hook/`.
- **Configure SCM in Jenkins Project:** The next step is to configure Git as a source code management tool within the Jenkins project by clicking on the *Configure* link on the project's dashboard. Following is a sample of input parameters:
  - Repository\_URL:  
`https://<bitbucket_user>@bitbucket.org/<bitbucket_user>/<bitbucket_project_name>.git`
  - Credentials: Create a credential by providing a BitBucket username and password.
- **Setup Build Triggers in Jenkins Project:** Finally, one would need to setup build triggers for the Jenkins project by clicking on the *Configure* link on project's dashboard. Check the option "Build when a change is pushed to BitBucket."

---

Site24x7 - Full stack It Infrastructure Monitoring from the cloud. Sign up for free trial.

---

## Like This Article? Read More From DZone



**DevOps Tools for Continuous Delivery in Jelastic Private Cloud. Part 2**



**Yelp's Distributed System for Concurrent Task Execution**




**Spinnaker: Netflix's New Cloud Deployment Automation Tool**



**Free DZone Refcard  
Getting To Know Google Compute Engine And How To Use It**

Topics: CLOUD, AWS EC2, JENKINS, BITBUCKET, CONTINUOUS DELIVERY

Published at DZone with permission of Ajitesh Kumar, DZone MVB. [See the original article here.](#)   
Opinions expressed by DZone contributors are their own.

---



