

Inicio » DevOps » Desarrolladores de Docker para Java: Desarrollar en Docker

SOBRE ANDREY REDKO



Andriy es un desarrollador de software bien fundamentado con más de 12 años de experiencia práctica utilizando Java / EE, C # / .NET, C ++, Groovy, Ruby, programación funcional (Scala), bases de datos (MySQL, PostgreSQL, Oracle) y soluciones NoSQL (MongoDB, Redis).



Desarrolladores de Docker para Java: Desarrollar en Docker

Publicado por: Andrey Redko en DevOps el 11 de diciembre de 2017

1. Introducción

En la última parte del tutorial , aprendimos varias formas de empaquetar (y ejecutar) aplicaciones Java como contenedores Docker . En esta parte nos enfocaremos más en cómo Docker impacta nuestras prácticas de desarrollo de una manera bastante positiva.

¿Quieres dominar a Docker?

¡Suscríbete a nuestro boletín y descarga Docker Containerization Cookbook ahora mismo!

Para ayudarlo a dominar Docker, ¡hemos compilado una guía increíble con todos los conceptos básicos del sistema de contenedores Docker! ¡Además de estudiarlos en línea, puedes descargar el eBook en formato PDF!

Dirección de correo electrónico:

Regístrate

Tabla de contenido

1. Introducción
2. Docker desde tu IDE
 - 2.1. Eclipse
 - 2.2. IntelliJ IDEA
 - 2.3. Oracle NetBeans
3. Al límite: tu IDE en Docker
4. Entonces, ¿qué pasa con la depuración?
5. Gestión y supervisión
 - 5.1. Portainer
 - 5.2. cAdvisor
 - 5.3. ctop
 - 5.4. Y mucho más ...
6. Docker Engine Metrics
7. Conclusiones
8. ¿Qué sigue?

Hablando en términos generales, hay muchas áreas donde Docker podría ser muy útil para los desarrolladores. El énfasis de nuestra discusión se centrará en la integración de Docker con IDE s, ejecutando IDE s como contenedores Docker y conociendo algunas herramientas útiles

HOJA INFORMATIVA

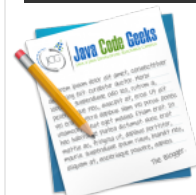
¡179.260 personas que información privilegiada disfrutamos actualizaciones semanales y **¡blancos de cortesía!** **Únete a ellos ahora** **acceso exclusivo** a las en el mundo de Java, así como sobre Android, Scala, Groovy tecnologías relacionadas.

Dirección de correo electrónico:

☒ Reciba alertas de trabajo desarrollador en su área

Regístrate

ÚNETE A NOSOTROS



Con **1,7** mensuales **500** artículos ubicados en sitios de Java. Cc buscamos animar a nosotros un blog con contenido único e in entonces debe consultar nuestro

2. Docker desde tu IDE

Como hemos visto hasta ahora, las herramientas de línea de comandos de Docker son muy potentes y su uso es bastante sencillo. Sin embargo, la mayoría de nosotros, desarrolladores, pasamos la mayor parte del día en nuestro IDE favorito . Cambiar entre docenas de ventanas y terminales en el medio de la sesión de depuración mientras busca un error difícil de reproducir en la lógica de la aplicación podría ser bastante contraproducente.

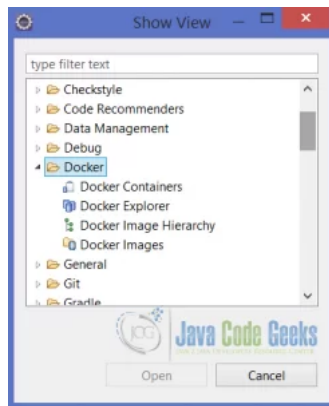
Afortunadamente, los tres principales IDE de desarrollo de Java , a saber, Eclipse , JetBrains IntelliJ IDEA y NetBeans , proporcionan integración con las herramientas de Docker a través de convenientes vistas de interfaz de usuario y / o widgets. Repasemos rápidamente cada uno de estos IDE y miremos más de cerca lo que tiene que ofrecer.

2.1. Eclipse

Eclipse Oxygen , la versión más reciente de Eclipse IDE a partir de este momento, cuenta con el completo soporte de herramientas Docker proporcionado por el subproyecto Linux Tools (disponible como un complemento). Una vez instalado, aporta cuatro vistas adicionales en la

Docker

categoría.



Vistas de Docker en Eclipse

Cada una de estas vistas expone algunos detalles sobre contenedores o imágenes de Docker , con

Docker Explorer

una vista integral de ambos.



Docker Explorer View

Si está interesado solo en contenedores Docker , hay una

Docker Containers

vista dedicada , que utiliza una representación compacta y basada en tablas.

Name	Image	Created	Command	Ports	Status
mysql	mysql:8.0.2	2017-11-19	docker-entrypoint.sh mysqld	0.0.0.0:3306->3306/tcp	Up 2 hours

Vista de Contenedores Docker

Del mismo modo, si desea enumerar solo imágenes de Docker , la

Docker Images

Repo Tags	Created	Virtual Size
10723c546a1	2017-11-09	133.9 MB
t80e0c3b8ac	2017-11-08	184.3 MB
5d006c8cfad1	2017-11-01	127.6 MB
b68295412d	2017-11-01	116.4 MB
3b18d34c52a	2017-10-25	101.1 MB
4be50ee404	2017-10-07	101 MB
69323205fa2	2017-09-19	266.4 MB

Vista de imágenes de Docker

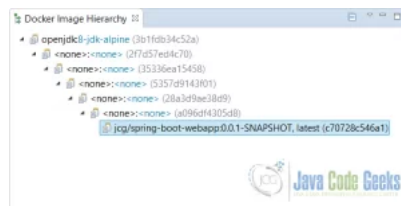
Por último, pero no menos importante, hay una vista muy útil llamada

Docker Image Hierarchy

. Permite ver todas las capas en las que se ha construido la imagen en cuestión. Aquí, por ejemplo, pudimos ver cómo la imagen etiquetada con el nombre

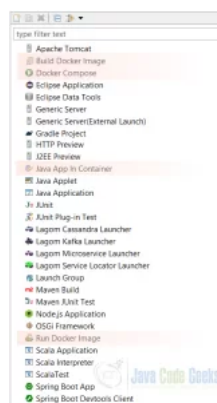
jcg/spring-boot-webapp:0.0.1-SNAPSHOT

(una de las muestras que estábamos construyendo en la parte previa del tutorial) está en capas.



Jerarquía de imágenes de Docker

Pero estos puntos de vista son solo la parte de la historia completa. Otra parte es un conjunto de nuevos lanzadores Docker que puede usar para construir imágenes Docker y ejecutar sus aplicaciones Java (o incluso completar pilas Docker Compose) como contenedores Docker .



Lanzadores de Eclipse

En caso de que desee obtener más información acerca de todas las vistas y lanzadores mencionados anteriormente, y, en general, sobre las herramientas Docker disponibles en Eclipse , el proyecto oficial Wiki es un excelente lugar para comenzar.

2.2. IntelliJ IDEA

El IDEA JetBrains IntelliJ, disponible en la versión más reciente

2017.2.6

, ofrece la Docker soporte de herramientas en una forma del plug-in oficial . Está disponible

Ultimate Edition

tanto como in

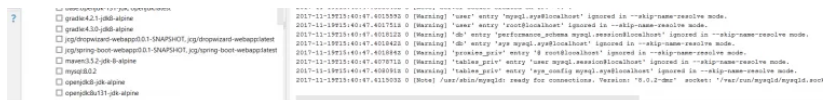
Community Edition

, que vamos a utilizar en esta sección.

Una vez que se instala el complemento, la nueva

Docker

vista estará disponible, con todos los detalles sobre las imágenes y los contenedores integrados.



Docker en IntelliJ Idea

Para ser justos, tener una única vista bien estructurada con la mayoría de las piezas requeridas disponibles en las cercanías (en las diferentes pestañas) es muy conveniente.

2.3. Oracle NetBeans

Los NetBeans desde la última

8.2

versión tiene acoplable de herramientas se incluye como parte del

Base IDE

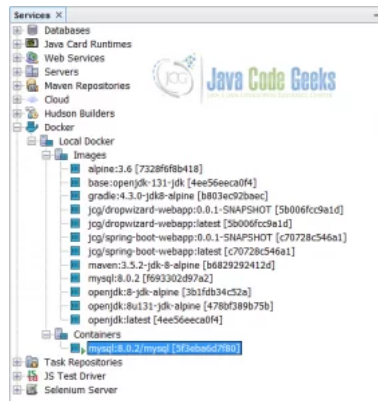
complemento (que es muy probable que le haya activado desde el principio). Después de la activación del complemento, el

Docker

nodo del árbol de adición queda disponible en la

Services

vista.



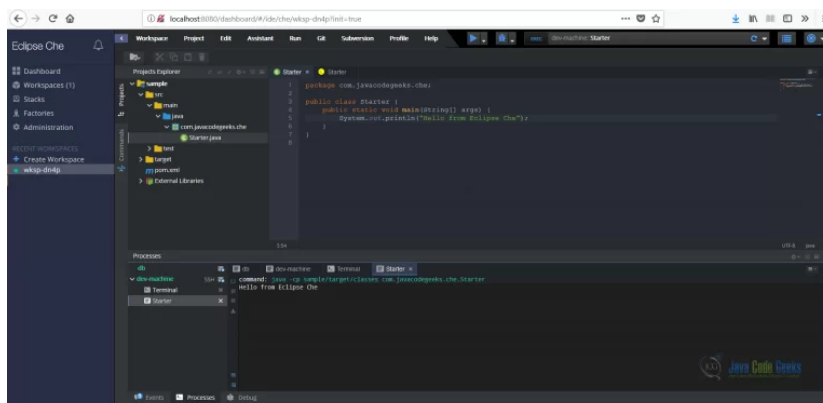
Docker en Netbeans

Un resumen bastante agradable y discusiones sobre las características deseadas con respecto a la integración de Docker en NetBeans se han publicado en Wiki de la comunidad oficial . Por favor, échale un vistazo si tienes curiosidad o estás interesado en este tema.

3. Al límite: tu IDE en Docker

Como hemos visto, las herramientas Docker proporcionadas por los IDE de desarrollo de Java más populares son bastante suficientes, se puede decir incluso sobresaliente. Pero, ¿qué pasa si tratamos de superar los límites y ejecutar nuestro IDE por completo en Docker , literalmente en cualquier lugar? Si suena un poco loco, seguramente sí, pero ...

Eclipse Che , el IDE de próxima generación de Eclipse , está aprovechando la contenedorización en toda su extensión y funciona completamente con Docker . Debajo del capó, está alimentado por REST (ful) API web y proporciona un IDE basado en navegador y rico en funciones .



Eclipse Che

4. Entonces, ¿qué pasa con la depuración?

La depuración fue, es y muy probablemente desempeñará un papel crucial en la búsqueda, comprensión y resolución de defectos o problemas en nuestros programas. Y, nos guste o no, no importa cuán asombroso sea Docker, no hará que nuestras aplicaciones estén libres de errores. Si está ejecutando sus aplicaciones Java como contenedores Docker, el proceso de depuración cambia un poco y se explica en gran detalle en este excelente artículo. Es tan completo que no hay necesidad de repetirlo aquí, solo elija el IDE que usa y repase los hermosos tutoriales preparados para usted.

5. Gestión y supervisión

Aunque las herramientas de línea de comandos de Docker y, en consecuencia, las API de Docker Engine, le permiten realizar cualquier tarea que pueda encontrar, muchos de nosotros todavía preferimos tratar con algún tipo de UI web o gráfica. Obviamente, esta es una preferencia personal y en esta sección aprenderemos sobre algunas opciones interesantes a este respecto.

5.1. Portainer

La primera herramienta que una va a tener en cuenta es Portainer, una interfaz de usuario de gestión de peso ligero de código abierto que permite gestionar fácilmente diferentes acoplables entornos (como Docker anfitriones o acoplable Swarm racimos). No sorprende que Portainer se ejecute como un contenedor Docker liviano. Vamos a ejecutar uno:

```
1 | $ docker run -d -p 9000:9000 -v /var/run/docker.sock:/var/run/docker.sock portainer/portainer
```

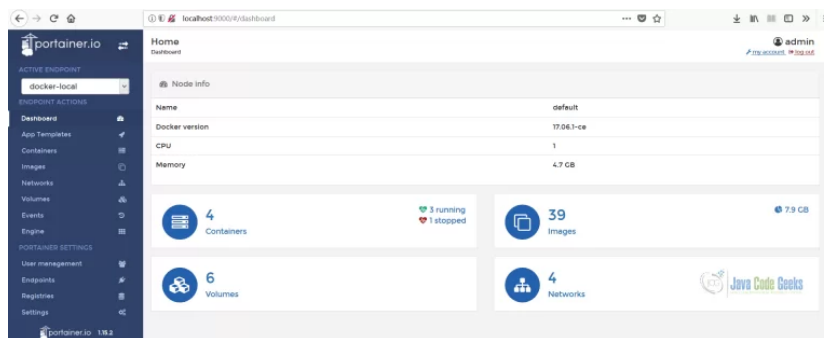
En la máquina de Windows, el comando se ve un poco diferente:

```
1 | $ docker run -d -p 9000:9000 portainer/portainer
```

Una vez que el contenedor Portainer esté en funcionamiento, podrá acceder a él a través de la interfaz de usuario web bastante ordenada disponible en <http://localhost:9000> (o, en caso de que esté utilizando una instalación Docker no nativa

```
http://<docker-machine>:9000
```

).



Portainer Dashboard

Cuando se completan los pasos de configuración (si es necesario), se lo dirigirá a la vista todo en uno del entorno Docker en cuestión, comenzando desde la página del tablero simple. Definitivamente reconocerá todos los objetos Docker (como volúmenes, redes, contenedores, imágenes, ...) enumerados a la izquierda para que pueda seleccionar cualquiera de ellos para ser redirigido a una vista dedicada.

La lista de funciones compatibles con Docker que Portainer expone a través de la interfaz web intuitiva es simplemente asombrosa. Si favorece las herramientas de UI sobre las de línea de comando, Portainer tiene todas las posibilidades de servir como la ventanilla única para usted.

5.2. cAdvisor

A menudo, mientras desarrollamos nuestras aplicaciones, es necesario obtener más información sobre lo que está sucediendo con nuestros contenedores, ya sea que hablemos de la CPU, la memoria o la utilización de la red. Afortunadamente, hay bastantes opciones aquí y una de las herramientas más útiles sobre las que vamos a hablar es cAdvisor (o en versión completa, asesor de contenedores) que analiza el uso de recursos y las características de rendimiento de los contenedores en ejecución.

“ cAdvisor brinda a los usuarios de contenedores una comprensión del uso de los recursos y las características de rendimiento de sus contenedores en funcionamiento. Es un daemon en ejecución que recopila, agrega, procesa y exporta información sobre contenedores en ejecución. Específicamente, para cada contenedor mantiene parámetros de aislamiento de recursos, uso histórico de recursos, histogramas de uso completo de recursos históricos y estadísticas de red. Esta información se exporta por contenedor y por toda la máquina. ”- <https://github.com/google/cadvisor>

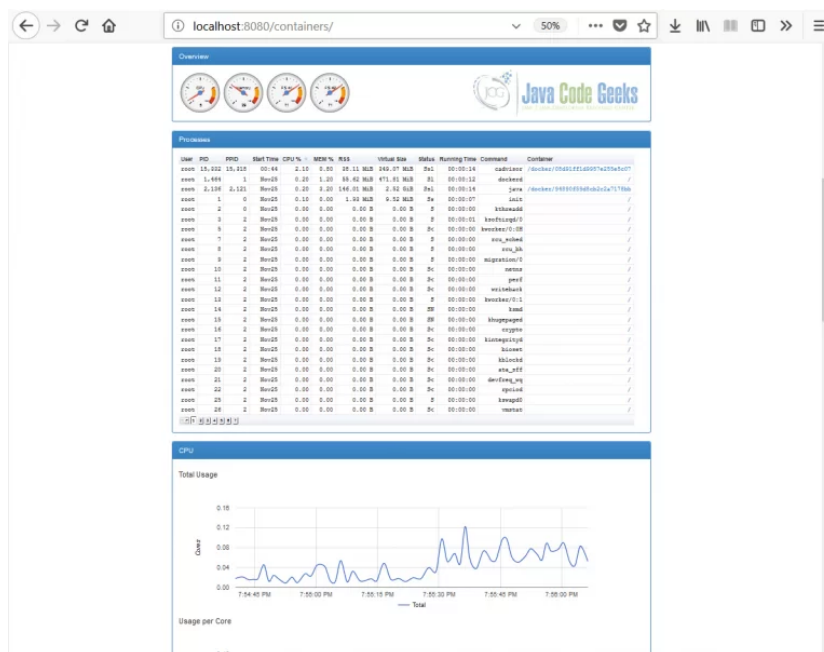
Curiosamente, cAdvisor va más allá de Docker y debería ser compatible con cualquier otro tipo de contenedor de forma inmediata. Tristemente para los usuarios de Windows, cAdvisor tiene soporte limitado en esta plataforma. La mejor manera de probar el cAdvisor es, sin duda, ejecutarlo como contenedor Docker.

```
05 --volume=/var/lib/docker:/var/lib/docker:ro \
06 --volume=/dev/disk:/dev/disk:ro \
07 --publish=8080:8080 \
08 --detach=true \
09 --name=cadvisor \
10 google/cadvisor:latest
```

Una vez que el contenedor esté en funcionamiento, puede acceder a la interfaz de usuario web de cAdvisor en <http://localhost:8080> (o, en caso de que esté utilizando una instalación Docker no nativa

http://<docker-machine>:8080

).

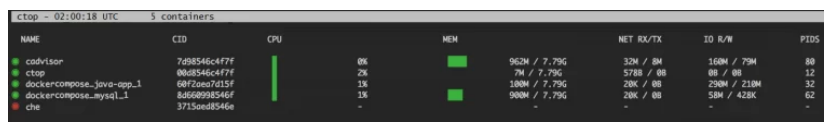


cAdvisor

La interfaz de usuario web es bastante simple pero excepcionalmente informativa. Se recopilan muchos detalles sobre los recursos y procesos de cada contenedor Docker en ejecución, así como sobre el propio equipo host Docker (CPU, memoria, red, almacenamiento, ...). De todos modos, es una herramienta invaluable para aprender cómo consumen sus contenedores y administrar los recursos en tiempo de ejecución.

5.3. ctop

Y por último, pero no menos importante, vamos a terminar con CTOP herramienta que, en una parte superior de interfaz -como, proporciona una visión concisa y condensada de métricas en tiempo real de varios contenedores.



ctop

No solo ctop tiene una vista de resumen, sino que también admite la opción de seleccionar y centrarse en cualquier contenedor individual.

5.4. Y mucho más ...

El espacio de herramientas Docker es realmente próspero, con nuevos proyectos interesantes apareciendo literalmente todos los días. Acabamos de cubrir un par de herramientas conocidas, maduras y útiles, pero hay mucho más por descubrir, así que por favor mire este espacio de cerca si está interesado.

6. Docker Engine Metrics

El motor Docker expone una gran cantidad de métricas interesantes en el formato que es totalmente compatible con Prometheus, la solución tremendamente popular de monitoreo y alerta de fuente abierta. Aunque esta característica se hizo disponible en general hace bastante tiempo, no mucha gente lo sabe y, como tal, no muchos la usan.

Esta característica todavía se considera experimental, por lo que no está disponible por defecto y necesita un poco de ajuste de configuración. La documentación oficial describe detalladamente todos los pasos necesarios a realizar con respecto al sistema operativo en el que está ejecutando Docker . Pero esencialmente, el primer paso es modificar

daemon.ison

```

2  "metrics-addr" : "localhost:9323",
3  "experimental" : true
4  }

```

Para que este cambio surta efecto, el daemon de Docker debe reiniciarse. Inmediatamente después, las métricas deberían estar disponibles en `http://localhost:9323/metrics` endpoint. El siguiente paso es preparar la configuración de Prometheus

```
prometheus.yml
```

para eliminar las métricas del motor Docker .

```

1 scrape_configs:
2   - job_name: 'prometheus'
3     static_configs:
4       - targets: ['localhost:9090']
5   - job_name: 'docker'
6     static_configs:
7       - targets: ['localhost:9323']

```

Con esta configuración mínima en mano, podríamos iniciar el servidor Prometheus (que ha alcanzado la versión

```
2.0
```

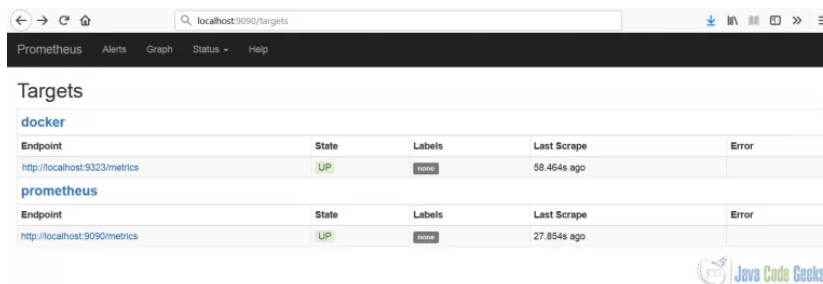
al momento de escribir esto) como contenedor Docker :

```

1 docker run -p 9090:9090 \
2   -v prometheus.yml:/etc/prometheus/prometheus.yml \
3   prom/Prometheus

```

Una vez que se inicia el contenedor, la interfaz de usuario web de Prometheus debe estar disponible en `http://localhost:9090` . Naveguemos a esta URL y verifique que todos los objetivos que hemos configurado estén accesibles.



Prometheus Targets

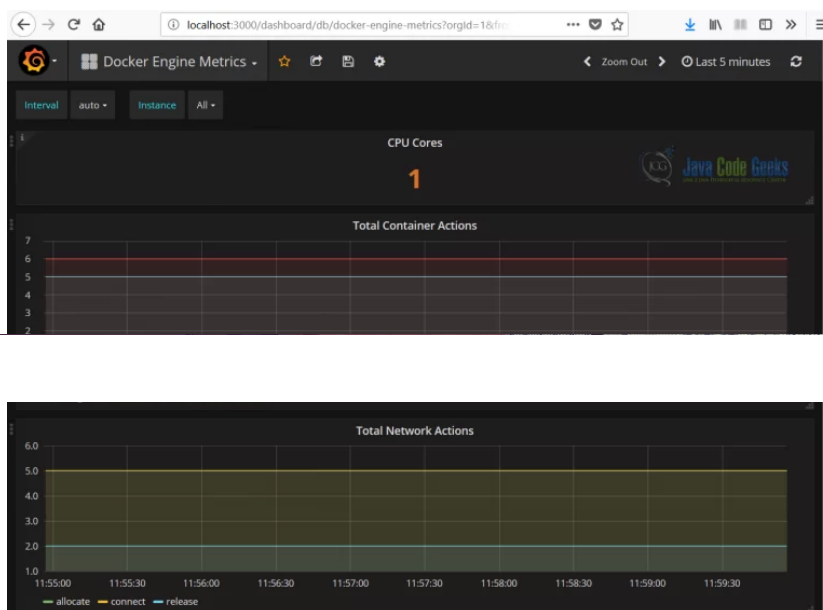
En términos generales, la interfaz de usuario web de Prometheus es bastante capaz de graficar diferentes métricas, pero Grafana , otro proyecto de código abierto, es la plataforma más avanzada para consultar, visualizar, alertar y comprender las métricas.

```

1 docker run -d -p 3000:3000 \
2   -e "GF_SECURITY_ADMIN_PASSWORD=pa$$w0rd" \
3   grafana/grafana

```

Grafana tiene una enorme lista de tableros preconstruidos y uno de ellos, el panel Docker Engine Metrics , es exactamente lo que estamos buscando. Aquí está el adelanto de cómo se muestra (hermosa, de hecho).



Tablero del motor Grafana Docker

7. Conclusiones

En esta sección del tutorial hemos hablado sobre cómo los mejores amigos del desarrollador, los IDE , se mantienen al tanto de los tiempos al proporcionar herramientas y soporte Docker . Más allá de eso, también hemos mencionado algunas herramientas muy útiles para administrar y monitorear sus contenedores y el propio motor Docker .

8. ¿Qué sigue?

En la siguiente sección del tutorial vamos a hablar sobre el uso de Docker para las pruebas, particularmente en el contexto de la realización de pruebas de integración y extremo a extremo de nuestras aplicaciones Java.

¿Desea saber cómo desarrollar sus habilidades para convertirse en Java Rockstar?



¡Suscríbete a nuestro boletín para comenzar a rocking

¡Para comenzar, te ofrecemos nuestros eBooks más vendidos

ahora mismo!

GRATIS!

1. Mini libro de JPA
2. Guía de solución de problemas de JVM
3. JUnit Tutorial para pruebas unitarias
4. Tutorial de anotaciones en Java
5. Preguntas de la entrevista de Java
6. Preguntas de la entrevista de primavera
7. Diseño de la interfaz de usuario de Android

y muchos más

Dirección de correo electrónico:

Regístrate

Etiquetado con: [DOCKER](#)

Deja una respuesta

¡Sé el primero en comentar!

Notificar de

Email

>



Start the discussion

[Java Code Geeks](#)
[Minilibros](#)
[Recursos](#)
[Tutoriales](#)

FOGONADURA

[Mkyong](#)

LA RED CODE GEEKS

[.NET Code Geeks](#)
[Java Code Geeks](#)
[Geeks del Código del Sistema](#)
[Geeks de código web](#)
[11 sitios web de aprendizaje en línea que debes visitar](#)
[Ventajas y desventajas de la computación en la nube - Pros y contras de computación en la nube](#)
[Tutorial de Android Google Maps](#)
[Análisis JSON de Android con Geon Tutorial](#)
[Aplicación de servicios basados en la ubicación de Android: ubicación del GPS](#)
[Tutorial de Preferencias rápidas de Android](#)
[Diferencia entre comparador y comparable en Java](#)
[GWT 2 Spring 3 JPA 2 Hibernate 3.5 Tutorial](#)
[Java Best Practices – Vector vs ArrayList vs HashSet](#)

JCGs serve the Java, SOA, Agile and Telecom communities with daily in domain experts, articles, tutorials, reviews, announcements, code snippets, source projects.

DISCLAIMER

Todas las marcas comerciales y marcas registradas que aparecen en Ja son propiedad de sus respectivos dueños. Java es una marca comercial registrada de Oracle Corporation en los Estados Unidos y otros países. J Code Geeks no está conectado a Oracle Corporation y no está patrocinado por Oracle Corporation.

