

Bonita

Tiempo estimado de lectura: 14 minutos

Bonita BPM es una suite de flujo de trabajo y gestión de procesos empresariales de código abierto

Repositorio de GitHub: https://github.com/Bonitasoft-Community/docker_bonita (https://github.com/Bonitasoft-Community/docker_bonita)

Referencia de la biblioteca

Este contenido se importa desde los documentos oficiales de Docker Library (<https://github.com/docker-library/docs/tree/master/bonita/>) y es proporcionado por el cargador original. Puede ver la página de Docker Store para este repo en <https://store.docker.com/images/bonita> (<https://store.docker.com/images/bonita>) .

Etiquetas compatibles y Dockerfile vínculos respectivos

- [7.3.3 \(7.3 / Dockerfile \)](https://github.com/Bonitasoft-Community/docker_bonita/blob/19d78ec0f212891e97814b2db30891b9b280f7e5/7.3/Dockerfile) (https://github.com/Bonitasoft-Community/docker_bonita/blob/19d78ec0f212891e97814b2db30891b9b280f7e5/7.3/Dockerfile)
- [7.4.3 , latest \(7.4 / Dockerfile \)](https://github.com/Bonitasoft-Community/docker_bonita/blob/5e4d4c6d86a90b2f7639215e4098097200a8751a/7.4/Dockerfile) (https://github.com/Bonitasoft-Community/docker_bonita/blob/5e4d4c6d86a90b2f7639215e4098097200a8751a/7.4/Dockerfile)

Para obtener información detallada sobre los artefactos publicados de cada una de las etiquetas compatibles anteriormente (metadatos de imagen, tamaño de transferencia, etc.), consulte el [repos/bonita](https://github.com/docker-library/repos/blob/master/repos/bonita) directorio (<https://github.com/docker-library/repos/blob/master/repos/bonita>) en el [docker-library/repo-info](https://github.com/docker-library/repo-info) reporte de GitHub (<https://github.com/docker-library/repo-info>) .

Para obtener más información acerca de esta imagen y su historial, consulte el archivo de manifiesto relevante ([library/bonita](https://github.com/docker-library/official-images/blob/master/library/bonita)) (<https://github.com/docker-library/official-images/blob/master/library/bonita>) . Esta imagen se actualiza a través de las solicitudes de extracción en el [docker-library/official-images](https://github.com/docker-library/official-images) reporte de GitHub (<https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fbonita>) .

¿Qué es Bonita BPM?

Bonita BPM es una suite de flujo de trabajo y gestión de procesos de negocio de código abierto creada en 2001. Se inició en el Instituto Nacional de Investigación en Informática de Francia y luego se incubó durante varios años dentro de la empresa informática francesa Groupe Bull. Desde 2009, el desarrollo de Bonita es apoyado por una empresa dedicada a esta actividad: Bonitasoft.

[Wikipedia.org/wiki/Bonita_BPM](http://en.wikipedia.org/wiki/Bonita_BPM) (http://en.wikipedia.org/wiki/Bonita_BPM)



Cómo usar esta imagen

Inicio rápido

```
$ docker run --name bonita -d -p 8080:8080 bonita
```

Esto iniciará un contenedor que ejecute el Tomcat Bundle (<http://documentation.bonitasoft.com/?page=tomcat-bundle>) con Bonita BPM Engine + Bonita BPM Portal. Sin especificar las variables de entorno, es como si hubieras lanzado el paquete en tu host mediante el inicio. {Sh

Bat} (con endurecimiento de seguridad en REST y API HTTP, cf Parte de seguridad). Bonita BPM utiliza una base de datos H2 aquí.

Puede acceder al Portal BPM de Bonita en <http://localhost:8080/> bonita y iniciar sesión usando las credenciales predeterminadas: install / install

Enlace Bonita BPM a una base de datos

PostgreSQL

PostgreSQL es la base de datos recomendada.

Establezca `max_prepared_transactions` en 100 (<http://documentation.bonitasoft.com/?page=database-configuration#toc4>) :

```
mkdir -p custom_postgres
echo '#!/bin/bash' > custom_postgres/bonita.sh
echo 'sed -i "s/^.*max_prepared_transactions\s*=\s*\(.*)$/max_prepared_transactions = 100/" "$PGDATA"/postgresql.conf' >> custom_postgres/bonita.sh
chmod +x custom_postgres/bonita.sh
```

Monte esa ubicación de directorio como `/docker-entrypoint-initdb.d` dentro del contenedor de PostgreSQL:

```
$ docker run --name mydbpostgres -v "$PWD"/custom_postgres:/docker-entrypoint-initdb.d -e POSTGRES_PASSWORD=mysecretpassword -
```

Consulte la documentación oficial de PostgreSQL (https://registry.hub.docker.com/_/postgres/) para obtener más detalles.

```
$ docker run --name bonita_postgres --link mydbpostgres:postgres -d -p 8080:8080 bonita
```

MySQL

Hay problemas conocidos con la administración de transacciones XA por el motor y el controlador de MySQL: vea los errores 17343 (<http://bugs.mysql.com/bug.php?id=17343>) y 12161 de (<http://bugs.mysql.com/bug.php?id=12161>) MySQL para más detalles. Por lo tanto, no se recomienda utilizar la base de datos MySQL en un entorno de producción.

Aumente el tamaño de paquete (<http://documentation.bonitasoft.com/?page=database-configuration#toc4>) que se establece de forma predeterminada en 1M:

```
mkdir -p custom_mysql
echo "[mysqld]" > custom_mysql/bonita.cnf
echo "max_allowed_packet=16M" >> custom_mysql/bonita.cnf
```

Monte esa ubicación del directorio como `/etc/mysql/conf.d` dentro del contenedor de MySQL:

```
$ docker run --name mydbmysql -v "$PWD"/custom_mysql:/etc/mysql/conf.d -e MYSQL_ROOT_PASSWORD=mysecretpassword -d mysql:5.5
```

Consulte la documentación oficial de MySQL (https://registry.hub.docker.com/_/mysql/) para obtener más detalles.

Inicie su contenedor de aplicaciones para vincularlo al contenedor MySQL:

```
$ docker run --name bonita_mysql --link mydbmysql:mysql -d -p 8080:8080 bonita
```

Modificar credenciales predeterminadas

```
$ docker run --name=bonita -e "TENANT_LOGIN=tech_user" -e "TENANT_PASSWORD=secret" -e "PLATFORM_LOGIN=pfadmin" -e "PLATFORM_PA
```

Ahora puedes acceder al Portal BPM de Bonita en localhost: 8080 / bonita y acceder con: tech_user / secret

Dónde almacenar datos

La mayoría de los datos se almacenan en una base de datos y se pueden almacenar fuera del contenedor Bonita como se describe anteriormente utilizando el contenedor PostgreSQL o MySQL. Sin embargo, algunos datos permanecen dentro del paquete Bonita. Bonita Home es una carpeta, llamada `bonita`, que contiene carpetas y archivos de configuración, de trabajo y temporales. También hay archivos de registro dentro de la `logs` carpeta.

Nota importante: Hay varias maneras de almacenar datos utilizados por las aplicaciones que se ejecutan en contenedores Docker. Animamos a los usuarios de las `bonita` imágenes a familiarizarse con las opciones disponibles, incluyendo:

- Permita que Docker administre el almacenamiento de sus datos escribiendo los archivos en el disco en el sistema host usando su propia gestión interna de volumen (<https://docs.docker.com/engine/tutorials/dockervolumes/#adding-a-data-volume>). Este es el valor predeterminado y es fácil y bastante transparente para el usuario. La desventaja es que los archivos pueden ser difíciles de localizar para herramientas y aplicaciones que se ejecutan directamente en el sistema host, es decir, fuera de los contenedores.
- Cree un directorio de datos en el sistema host (fuera del contenedor) y monte esto en un directorio visible desde el interior del contenedor (<https://docs.docker.com/engine/tutorials/dockervolumes/#mount-a-host-directory-as-a-data-volume>). Esto coloca los

archivos de base de datos en una ubicación conocida en el sistema host y facilita que las herramientas y las aplicaciones del sistema host accedan a los archivos. La desventaja es que el usuario necesita asegurarse de que el directorio existe y que los permisos de directorio y otros mecanismos de seguridad en el sistema host están configurados correctamente.

La documentación de Docker es un buen punto de partida para entender las diferentes opciones de almacenamiento y las variaciones, y hay varios blogs y publicaciones de foro que discuten y dan consejos en esta área. Simplemente mostraremos el procedimiento básico aquí para la última opción arriba:

1. Cree un directorio de datos en un volumen adecuado en su sistema host, por ejemplo `/my/own/datadir`.
2. Inicie su `bonita` contenedor de esta manera:

```
Docker run -name algunos-bonita -v /mi / propio / datadir: / opt / bonita -d bonita: tag
```

La `-v /my/own/datadir:/opt/bonita` parte del comando monta el `/my/own/datadir` directorio del sistema host subyacente como `/opt/bonita` dentro del contenedor, donde Bonita desplegará el paquete y escribirá los archivos de datos de forma predeterminada.

Tenga en cuenta que los usuarios de sistemas host con SELinux habilitado pueden ver problemas con esto. La solución actual es asignar el tipo de política SELinux relevante al nuevo directorio de datos para que el contenedor pueda acceder a él:

```
$ chcon -Rt svirt_sandbox_file_t /my/own/datadir
```

Migrar desde una versión anterior de Bonita BPM

- Detener el contenedor para realizar una copia de seguridad

```
$ docker stop bonita_7.2.3_postgres
```

- Para contenedores <7.3.0:
 - Compruebe dónde se almacenan sus datos

```
$ docker inspect bonita_7.2.3_postgres
[...]
```

```
  "Mounts": [
    {
      "Source": "/home/user/Documents/Docker/Volumes/bonita_7.2.3_postgres",
      "Destination": "/opt/bonita",
      "Mode": "",
      "RW": true
    }
  ],
[...]
```

- Copiar datos desde el sistema de archivos

```
cp -r bonita_7.2.3_postgres bonita_migration
```

- Recuperar el IP del contenedor de DB

```
$ docker inspect --format '{{ mydbpostgres
172.17.0.26
```

- Borrar la base de datos

```
export PGPASSWORD=mysecretpassword
pg_dump -O -x -h 172.17.0.26 -U postgres bonitadb > /tmp/bonitadb.sql
```

Tenga en cuenta que businessdb no se actualizará con la herramienta de migración, pero es posible que también desee copiarla / moverla.

- Cargar el turgurio

```
export PGPASSWORD=mysecretpassword
psql -U postgres -h 172.17.0.26 -d postgres -c "CREATE USER newbonitauser WITH PASSWORD 'newbonitapass';"
psql -U postgres -h 172.17.0.26 -d postgres -c "CREATE DATABASE newbonitadb OWNER newbonitauser;"
export PGPASSWORD=newbonitapass
cat /tmp/bonitadb.sql | psql -U newbonitauser -h 172.17.0.26 newbonitadb
```

- Recuperar la última herramienta de migración

- Si migra a una versión <7.3.0

- Obtener también la versión de destino del paquete Bonita

```
cd bonita_migration
wget http://download.forge.ow2.org/bonita/bonita-migration-distrib-2.20.0.zip
wget http://download.forge.ow2.org/bonita/BonitaBPMCommunity-7.2.4-Tomcat-7.0.67.zip
unzip bonita-migration-distrib-2.20.0.zip
unzip BonitaBPMCommunity-7.2.4-Tomcat-7.0.67.zip
```

- Mover el Hogar anterior al nuevo paquete

```
mv BonitaBPMCommunity-7.2.4-Tomcat-7.0.67/bonita/ BonitaBPMCommunity-7.2.4-Tomcat-7.0.67/bonita.orig
cp -r BonitaBPMCommunity-7.2.3-Tomcat-7.0.67/bonita/ BonitaBPMCommunity-7.2.4-Tomcat-7.0.67/bonita/
```

- Si migra a una versión >= 7.3.0

```
cd bonita_migration
wget http://download.forge.ow2.org/bonita/bonita-migration-distrib-2.20.0.zip
unzip bonita-migration-distrib-2.20.0.zip
```

- Configurar la herramienta de migración

```
cd bonita-migration-distrib-2.20.0
```

Editar la configuración de la herramienta de migración para apuntar hacia la copia de bonita casa y db

```
vim Config.properties
```

Por ejemplo :

```
db.vendor=postgres
db.url=jdbc:postgresql://172.17.0.26:5432/newbonitadb
db.driverClass=org.postgresql.Driver
db.user=newbonitauser
db.password=newbonitapass
# location of the bonita home (only useful when migration from version before 7.3.0)
bonita.home=/home/user/Documents/Docker/Volumes/bonita_migration/BonitaBPMCommunity-7.2.3-Tomcat-7.0.67/bonita
```

- Iniciar la migración

```
cd bin
./bonita-migration-distrib
```

- Lanzar el nuevo contenedor apuntando hacia la copia de la base de datos y el sistema de archivos

- Si <7.3.0

```
$ docker run --name=bonita_7.2.4_postgres --link mydbpostgres:postgres -e "DB_NAME=newbonitadb" -e "DB_USER=newbonitauser"
```



- Si >= 7.3.0

```
$ docker run --name=bonita_7.4.2_postgres --link mydbpostgres:postgres -e "DB_NAME=newbonitadb" -e "DB_USER=newbonitauser"
```



- Vuelva a aplicar la configuración específica si es necesario, por ejemplo con una versión >= 7.3.0:

```
$ docker exec -ti bonita_7.4.2_postgres /bin/bash
```

```
$ cd /opt/bonita/BonitaBPMCommunity-7.4.2-Tomcat-7.0.67/setup
$ ./setup.sh pull
$ TENANT_LOGIN=tech_user
$ TENANT_PASSWORD=secret
$ PLATFORM_LOGIN=pfadmin
$ PLATFORM_PASSWORD=pfsecret
$ sed -e 's/^#userName\s*=.*/'"userName=${TENANT_LOGIN}"'/ ' \
    -e 's/^#userPassword\s*=.*/'"userPassword=${TENANT_PASSWORD}"'/ ' \
    -i platform_conf/current/tenants/1/tenant_engine/bonita-tenant-community-custom.properties
$ sed -e 's/^#platform.tenant.default.username\s*=.*/'"platform.tenant.default.username=${TENANT_LOGIN}"'/ ' \
    -e 's/^#platform.tenant.default.password\s*=.*/'"platform.tenant.default.password=${TENANT_PASSWORD}"'/ ' \
    -i platform_conf/current/platform_portal/platform-tenant-config.properties
$ sed -e 's/^#platformAdminUsername\s*=.*/'"platformAdminUsername=${PLATFORM_LOGIN}"'/ ' \
    -e 's/^#platformAdminPassword\s*=.*/'"platformAdminPassword=${PLATFORM_PASSWORD}"'/ ' \
    -i platform_conf/current/platform_engine/bonita-platform-community-custom.properties
$ sed -i -e 's/^#GET|/GET|/' -e 's/^#POST|/POST|/' -e 's/^#PUT|/PUT|/' -e 's/^#DELETE|/DELETE|/' -i platform_conf/current
$ ./setup.sh push
```

```
$ docker restart bonita_7.4.2_postgres
```

Para obtener más detalles sobre la migración de Bonita, consulte la documentación (<http://documentation.bonitasoft.com/?page=migrate-from-an-earlier-version-of-bonita-bpm>) .

Seguridad

Esta imagen Docker activa las comprobaciones de autorización estáticas y dinámicas de forma predeterminada en la API REST. Para ser coherente, también desactiva la API HTTP.

- Autorización REST API
 - Comprobación de autorización estática (<http://documentation.bonitasoft.com/?page=rest-api-authorization#toc1>)
 - Comprobación de autorización dinámica (<http://documentation.bonitasoft.com/?page=rest-api-authorization#toc2>)
- API HTTP (<http://documentation.bonitasoft.com/?page=rest-api-authorization#toc9>)

Para necesidades específicas, puede anular este comportamiento estableciendo HTTP_API en true y REST_API_DYN_AUTH_CHECKS en false:

```
$ docker run -e HTTP_API=true -e REST_API_DYN_AUTH_CHECKS=false --name bonita -d -p 8080:8080 bonita
```

Variables de entorno

Al iniciar la `bonita` imagen, puede ajustar la configuración de la instancia de Bonita pasando una o más variables de entorno en la `docker run` línea de comandos.

PLATFORM_PASSWORD

Esta variable de entorno se recomienda (<http://documentation.bonitasoft.com/?page=tomcat-bundle#toc3>) para que utilice la imagen de Bonita. Establece la contraseña del administrador de la plataforma para Bonita. Si no se especifica, `platform` se utilizará la contraseña predeterminada .

PLATFORM_LOGIN

Esta variable de entorno opcional se utiliza junto con `PLATFORM_PASSWORD` para definir el nombre de usuario para el administrador de la plataforma. Si no se especifica, `platformAdmin` se utilizará el usuario predeterminado .

TENANT_PASSWORD

Esta variable de entorno se recomienda (<http://documentation.bonitasoft.com/?page=tomcat-bundle#toc3>) para que utilice la imagen de Bonita. Establece la contraseña del administrador del inquilino para Bonita. Si no se especifica, `install` se utilizará la contraseña predeterminada .

TENANT_LOGIN

Esta variable de entorno opcional se utiliza junto con `TENANT_PASSWORD` para definir el nombre de usuario para el administrador del inquilino. Si no se especifica, `install` se utilizará el usuario predeterminado de .

REST_API_DYN_AUTH_CHECKS

Esta variable de entorno opcional se utiliza para habilitar / deshabilitar la comprobación de autorización dinámica (<http://documentation.bonitasoft.com/?page=rest-api-authorization#toc2>) en la API Bonita REST. El valor predeterminado es `true` , que activará la comprobación de autorización dinámica.

HTTP_API

Esta variable de entorno opcional se utiliza para habilitar / deshabilitar la API Bonita HTTP. El valor predeterminado es `false` , que desactivará la API HTTP.

JAVA_OPTS

Esta variable de entorno opcional se utiliza para personalizar JAVA_OPTS. El valor predeterminado es

`-Xms1024m -Xmx1024m -XX:MaxPermSize=256m` .

ENSURE_DB_CHECK_AND_CREATION

Esta variable de entorno opcional se utiliza para permitir / no permitir que las consultas SQL comprueben y creen automáticamente las bases de datos con las credenciales de administrador de la base de datos. El valor predeterminado es `true` .

DB_VENDOR

Esta variable de entorno se establece automáticamente en `postgres` o `mysql` si el contenedor Bonita está vinculado a una base de datos PostgreSQL o MySQL utilizando `--link` . El valor predeterminado es `h2` . Puede ser anulada si no utiliza la `--link` capacidad.

DB_HOST , DB_PORT

Estas variables son opcionales, utilizadas conjuntamente para configurar la `bonita` imagen para llegar a la instancia de la base de datos. Se establece automáticamente si `--link` se utiliza para ejecutar el contenedor.

DB_NAME , DB_USER , DB_PASS

Estas variables se utilizan conjuntamente para crear un nuevo usuario, establecer la contraseña del usuario y crear la `bonita` base de datos.

`DB_NAME` El valor predeterminado es `bonitadb` .

`DB_USER` El valor predeterminado es `bonitauser` .

`DB_PASS` El valor predeterminado es `bonitapass` .

BIZ_DB_NAME , BIZ_DB_USER , BIZ_DB_PASS

Estas variables se utilizan conjuntamente para crear un nuevo usuario, establecer la contraseña del usuario y crear la `bonita` base de datos empresarial (<http://documentation.bonitasoft.com/?page=define-and-deploy-the-bdm#toc1>) .

`BIZ_DB_NAME` El valor predeterminado es `businessdb` .

`BIZ_DB_USER` El valor predeterminado es `businessuser` .

`BIZ_DB_PASS` El valor predeterminado es `businesspass` .

DB_ADMIN_USER , DB_ADMIN_PASS

Estas variables son opcionales y se utilizan conjuntamente para crear usuarios y bases de datos a través de la cuenta de administrador utilizada en la instancia de la base de datos.

`DB_ADMIN_USER` Si no se proporciona ningún valor, se establece automáticamente `root` con MySQL o `postgres` con PostgreSQL.

`DB_ADMIN_PASS` Si no se proporciona ningún valor, se establece automáticamente mediante el valor del contenedor vinculado:

`MYSQL_ENV_MYSQL_ROOT_PASSWORD` o `POSTGRES_ENV_POSTGRES_PASSWORD` .

Cómo ampliar esta imagen

Si desea realizar una inicialización adicional, puede agregar un `*.sh` script en `/opt/custom-init.d` . El `startup.sh` archivo generará cualquier `*.sh` script que se encuentre en este directorio para realizar una inicialización posterior antes de iniciar el servicio.

Por ejemplo, puede aumentar el nivel de registro:

```
mkdir -p custom_bonita
echo '#!/bin/bash' > custom_bonita/bonita.sh
echo 'sed -i "s/^org.bonitasoft.level = WARNING$/org.bonitasoft.level = FINEST/" /opt/bonita/BonitaBPMCommunity-7.4.2-Tomcat-7.'
chmod +x custom_bonita/bonita.sh

docker run --name bonita_custom -v "$PWD"/custom_bonita:/opt/custom-init.d -d -p 8080:8080 bonita
```

Nota: Hay varias maneras de comprobar los `bonita` registros. Uno de ellos es

```
$ docker exec -ti bonita_custom /bin/bash
tail -f /opt/bonita/BonitaBPMCommunity-7.4.2-Tomcat-7.0.67/server/logs/bonita.`date +%Y-%m-%d`.log
```

Licencia

La imagen de Bonita BPM incluye dos partes:

- Bonita BPM Engine bajo LGPL v2.1 (<https://www.gnu.org/licenses/old-licenses/lgpl-2.1.html>)
- Bonita BPM Portal bajo GPL v2.0 (<http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>)

Versiones Docker soportadas

Esta imagen es oficialmente compatible con Docker versión 17.04.0-ce.

Soporte para versiones anteriores (hasta 1.6) se proporciona sobre una base de mejor esfuerzo.

Consulte la documentación de instalación de Docker (<https://docs.docker.com/installation/>) para obtener detalles sobre cómo actualizar el daemon de Docker.

Comentarios de los usuarios

Cuestiones

Si tiene algún problema o duda sobre esta imagen, póngase en contacto con nosotros a través de un problema de GitHub (https://github.com/Bonitasoft-Community/docker_bonita/issues) . Si el problema está relacionado con un CVE, compruebe primero (<https://github.com/docker-library/official-images/issues?q=label%3Acve-tracker>) un `cve-tracker` problema en el `official-images` repositorio (<https://github.com/docker-library/official-images/issues?q=label%3Acve-tracker>) .

También puede acceder a muchos de los mantenedores de imágenes oficiales a través del `#docker-library` canal IRC en Freenode (<https://freenode.net>) .

Contribuyendo

Se le invita a aportar nuevas características, arreglos o actualizaciones, grandes o pequeñas; Siempre estamos encantados de recibir solicitudes de tracción, y hacemos nuestro mejor esfuerzo para procesarlos tan rápido como podamos.

Antes de empezar a codificar, le recomendamos que discuta sus planes a través de un problema de GitHub (https://github.com/Bonitasoft-Community/docker_bonita/issues) , especialmente para contribuciones más ambiciosas. Esto le da a otros contribuyentes la oportunidad de señalarle en la dirección correcta, darle retroalimentación sobre su diseño y ayudarlo a averiguar si alguien más está trabajando en lo mismo.

Documentación

La documentación de esta imagen se almacena en el `bonita/` directorio (<https://github.com/docker-library/docs/tree/master/bonita>) del `docker-library/docs` repo GitHub (<https://github.com/docker-library/docs>) . Asegúrese de familiarizarse con el archivo (<https://github.com/docker-library/docs/blob/master/README.md>) del repositorio `README.md` (<https://github.com/docker-library/docs/blob/master/README.md>) antes de intentar una solicitud de extracción.

🔍 Biblioteca (<https://docs.docker.com/glossary/?term=library>) , muestra (<https://docs.docker.com/glossary/?term=sample>) , Bonita (<https://docs.docker.com/glossary/?term=Bonita>)

Califica esta página:

0 0

¿Qué es Docker? (<https://www.docker.com/what-docker>)

¿Qué es un contenedor? (<https://www.docker.com/what-container>)

Casos de Uso (<https://www.docker.com/use-cases>)

Clientes (<https://www.docker.com/customers>)

Fogonadura (<https://www.docker.com/partners/partner-program>)

Para el gobierno (<https://www.docker.com/industry-government>)

Acerca de Docker (<https://www.docker.com/company>)

administración (<https://www.docker.com/company/management>)

Prensa y Noticias (<https://www.docker.com/company/news-and-press>)

Empleo (<https://www.docker.com/careers>)

Producto (<https://www.docker.com/products/overview>)

Precio (<https://www.docker.com/pricing>)

Edición de comunidad (<https://www.docker.com/docker-community>)

Edición de Empresa (<https://www.docker.com/enterprise>)

Docker Datacenter (<https://www.docker.com/products/docker-datacenter>)

Docker Cloud (<https://cloud.docker.com/>)

Tienda Docker (<https://store.docker.com/>)

Docker para Mac (<https://www.docker.com/docker-mac>)

Docker para Windows (PC) (<https://www.docker.com/docker-windows>)

Docker para AWS (<https://www.docker.com/docker-aws>)

Docker para Azure (<https://www.docker.com/docker-microsoft-azure>)

Docker para Windows Server (<https://www.docker.com/docker-windows-server>)

Docker para distribución CentOS (<https://www.docker.com/docker-centos>)

Docker para Debian (<https://www.docker.com/docker-debian>)

Docker para Fedora® (<https://www.docker.com/docker-fedora>)

Docker para Oracle Enterprise Linux (<https://www.docker.com/docker-oracle-linux>)

Docker para RHEL (<https://www.docker.com/docker-rhel>)

Docker para SLES (<https://www.docker.com/docker-sles>)

Docker para Ubuntu (<https://www.docker.com//docker-ubuntu>)

Documentación (/)

Aprender (<https://www.docker.com/docker>)

Blog (<https://blog.docker.com>)

Formación (<https://training.docker.com/>)

Apoyo (<https://www.docker.com/docker-support-services>)

Base de conocimientos (<https://success.docker.com/kbase>)

Recursos (<https://www.docker.com/products/resources>)

Comunidad (<https://www.docker.com/docker-community>)

- Fuente abierta (<https://www.docker.com/technologies/overview>)
- Eventos (<https://www.docker.com/community/events>)
- Foros (<https://forums.docker.com/>)
- Capitanes del muelle (<https://www.docker.com/community/docker-captains>)
- Becas (<https://www.docker.com/docker-community/scholarships>)
- Noticias de la comunidad (<https://blog.docker.com/curated/>)

- Estado (<http://status.docker.com/>) Seguridad (<https://www.docker.com/docker-security>) Legal (<https://www.docker.com/legal>)
- Contacto (<https://www.docker.com/company/contact>)

Copyright © 2017 Docker Inc. Todos los derechos reservados.



