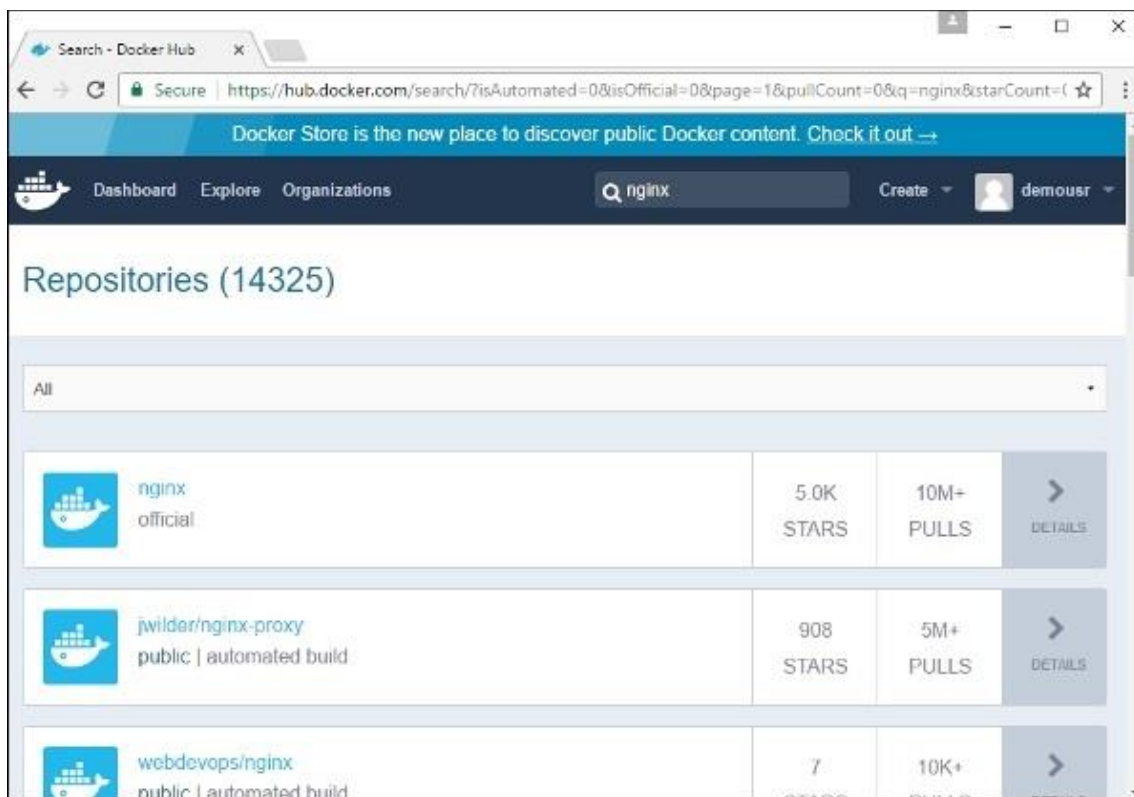


Docker - Configuración de NGINX

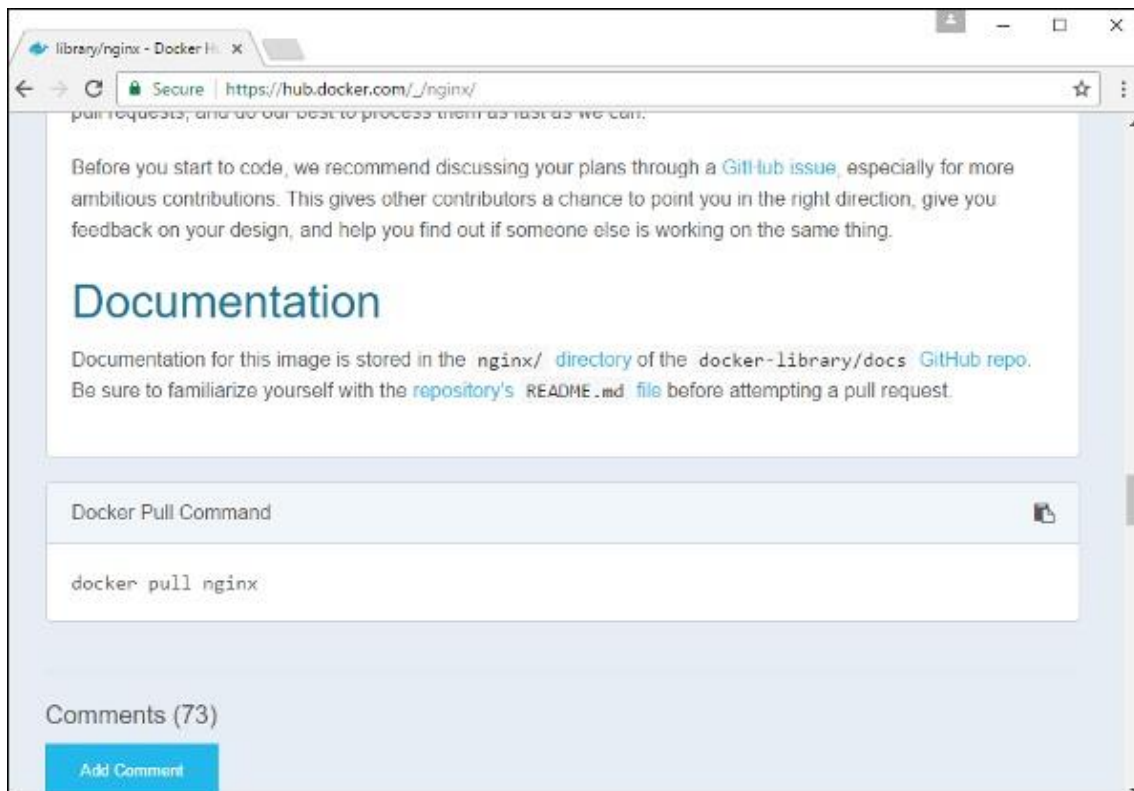
NGINX es una popular aplicación web ligera que se utiliza para desarrollar aplicaciones del lado del servidor. Es un servidor web de código abierto desarrollado para ejecutarse en una variedad de sistemas operativos. Dado que **nginx** es un servidor web popular para el desarrollo, Docker se ha asegurado de que tenga soporte para **nginx**.

Ahora veremos los distintos pasos para poner en funcionamiento el contenedor Docker para **nginx**.

Paso 1 : el primer paso es extraer la imagen de Docker Hub. Cuando inicie sesión en Docker Hub, podrá buscar y ver la imagen de **nginx** como se muestra a continuación. Simplemente escriba nginx en el cuadro de búsqueda y haga clic en el enlace **nginx** (oficial) que aparece en los resultados de búsqueda.



Paso 2 : verá que el comando de **extracción de Docker** para **nginx** en los detalles del repositorio en Docker Hub.



Paso 3 : en el host de Docker, use el comando de **extracción de** Docker como se muestra arriba para descargar la última imagen nginx de Docker Hub.

```
demo@ubuntudemo:~$ sudo docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx

75a822cd7888: Already exists
0aefb9dc4a57: Pull complete
046e44ee6057: Pull complete
Digest: sha256:fab482910aae9630c93bd24fc6fcec9f9f792c24a8974f5e46d8ad625ac235
Status: Downloaded newer image for nginx:latest
demo@ubuntudemo:~$ _
```

Paso 4 : ahora **ejecutemos** el contenedor **nginx** mediante el siguiente comando.

```
sudo docker run -p 8080:80 -d nginx
```

Estamos exponiendo el puerto en el servidor **nginx** que es el puerto 80 al puerto 8080 en el Docker Host.

```
demo@ubuntudemo:~$ sudo docker run -p 8080:80 -d nginx
[sudo] password for demo:
605d088b4d044f40b558fd321df2b40cadfcd5d0c59a947bddffffdc0c5151e4f
demo@ubuntudemo:~$
```

Una vez que ejecute el comando, obtendrá el siguiente resultado si navega a la URL **http://dockerhost:8080** . Esto muestra que el contenedor **nginx** está en funcionamiento.



Paso 5 : veamos otro ejemplo en el que podemos alojar una página web simple en nuestro contenedor **nginx** . En nuestro ejemplo, crearemos un archivo **HelloWorld.html** simple y lo **alojaremos** en nuestro contenedor **nginx** .

Primero creemos un archivo HTML llamado **HelloWorld.html**

```
demo@ubuntudemo:~$ sudo vim HelloWorld.html_
```

Agreguemos una línea simple de Hello World en el archivo HTML.

```
Hello World
```

Entonces, ejecutemos el siguiente comando de Docker.

```
sudo docker run -p 8080:80 -v  
"$PWD":/usr/share/nginx/html:ro -d nginx
```

Los siguientes puntos deben tenerse en cuenta sobre el comando anterior:

- Estamos exponiendo el puerto en el servidor **nginx** que es el puerto 80 al puerto 8080 en el Docker Host.
- A continuación, adjuntamos el volumen en el contenedor que es **/usr / share / nginx / html** a nuestro directorio de trabajo actual. Aquí es donde se almacena nuestro archivo HelloWorld.html.

```
demo@ubuntu:~$ sudo docker run -p 8080:80 -v "$PWD":/usr/share/nginx/html:rw -d nginx
bee9ef3fb04f2f3a219fc60123989351c63d4074a32a765340491f28ad3dbc44
demo@ubuntu:~$
demo@ubuntu:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS            PORTS              NAMES
bee9ef3fb04f        nginx              "nginx -g 'daemon off'" About a minute
ago               Up 50 seconds      443/tcp, 0.0.0.0:8080->80/tcp agitated_leavitt
ec086eec7416        mongo             "/entrypoint.sh mongo" About an hour
ago               Up About an hour   27017/tcp              tender_poitras
demo@ubuntu:~$ _
```

Ahora, si **navegamos** a la URL **http: // dockerhost: 8080 / HelloWorld.html** , obtendremos el siguiente resultado como se esperaba:

