

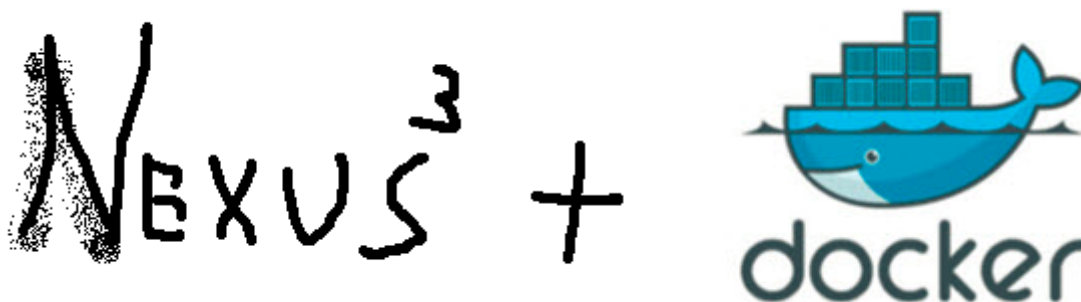


Crear un registro de Docker privado

Por Ivan Krizsan | 9 de junio de 2016

6 Comentarios

Contenido [[show](#)]



Nexus 3 y Docker - No pude usar el logotipo Nexus 3, así que dibujé uno yo mismo!

En este artículo voy a mostrar cómo configurar un registro de Docker privado con [Nexus Repository Manager 3.0 OSS](#) de Sonatype . Puede haber otras guías para esto, pero si nada más estoy escribiendo para mí. Las principales motivaciones para que yo establezca un registro privado de Docker son:

- Distribución de imágenes de Docker.
Esto puede ser a pequeña escala entre varios ordenadores o dentro de una organización.
- Restringir la distribución de las imágenes de Docker.
Aunque prefiero que mis imágenes de Docker estén disponibles en [DockerHub](#) , hay casos en los que siento que las imágenes de Docker no están listas para su distribución a una audiencia más grande. En una organización, puede ser indeseable o inadecuado implementar imágenes de Docker en un registro público.
- Utilice versiones idénticas de las imágenes de Docker en todos mis equipos.
Si construyo una imagen Docker en la primera computadora y luego, en algún momento posterior, en otra computadora, puede haber diferencias entre las imágenes. Esto puede deberse a actualizaciones de los repositorios de software utilizados al crear las imágenes, parches al sistema operativo, etc. Tales diferencias pueden ser sutiles, pero prefiero eliminar tantas posibles fuentes de confusión, o incluso errores, que posiblemente pueda.

Ya que ya estoy jugando con Docker, voy a ejecutar Nexus 3 OSS en un contenedor Docker utilizando la [imagen Docker de Sonatype](#).

Preparativos

Como ejemplo, haré una copia de la imagen oficial de Tomcat Docker en mi propio registro privado de Docker. Para ello, primero debemos sacar la imagen de Tomcat:

```
1 docker pull tomcat:latest
```

Además, crearé un directorio en el host para almacenar los datos y registros de Nexus 3, como se describe en la documentación de la imagen del Nexus 3:

```
1 mkdir -p /some/dir/nexus3/data
2 chown -R 200 /some/dir/nexus3/data
```

Crear una configuración de Docker Compose

Para simplificar el inicio del contenedor Nexus 3 Docker, utilizaré Docker Compose.

- Vaya al directorio "nexus3".
Este es el directorio padre del directorio "data" que creamos como parte de los preparativos anteriores.
- Cree un archivo denominado "docker-compose.yml" con el siguiente contenido:

```
1 nexus:
2   image: sonatype/nexus3:latest
3   ports:
4     - "8081:8081"
5     - "8123:8123"
6   volumes:
7     - ./data:/nexus-data
```

Tenga en cuenta que:

- Dos puertos están expuestos.
En el puerto 8081 estará disponible la GUI web de Nexus 3. La API del Registro de Docker estará expuesta en el puerto 8123. Este último puerto puede cambiarse como se desee, solo recuerde configurar Nexus 3 en consecuencia.
- El directorio compartido "nexus-data", que se asigna al directorio del host "data".
Este directorio contendrá los datos persistentes de Nexus 3.

Iniciar Nexus

En una ventana de terminal, el terminal Docker Quickstart si está utilizando Docker Toolbox, con el directorio actual siendo el directorio con el archivo docker-compose.yml que creamos anteriormente, inicie el administrador del Repositorio Nexus 3 usando este comando:

```
1 docker-compose up -d
```

Si está ejecutando Docker en Linux, tendrá que agregar "sudo" antes del comando anterior. A continuación, utilice el comando logs para ver los registros del contenedor Nexus Docker:

```
1 docker-compose logs
```

Después de algún tiempo, una salida de registro similar a esto debería aparecer en el registro de Nexus, lo que indica que Nexus ha iniciado correctamente:

```
1 nexus_1 | -----
2 nexus_1 |
3 nexus_1 | Started Sonatype Nexus OSS 3.0.0-03
4 nexus_1 |
5 nexus_1 | -----
```

Crear un registro de Docker en Nexus 3

Con Nexus 3 en funcionamiento, ahora deberíamos ser capaces de crear un registro de Docker.

- Busque la dirección IP del host Docker.

Si está ejecutando Docker en Mac OS X o en Windows, se puede obtener mediante el comando "docker-machine ip default".

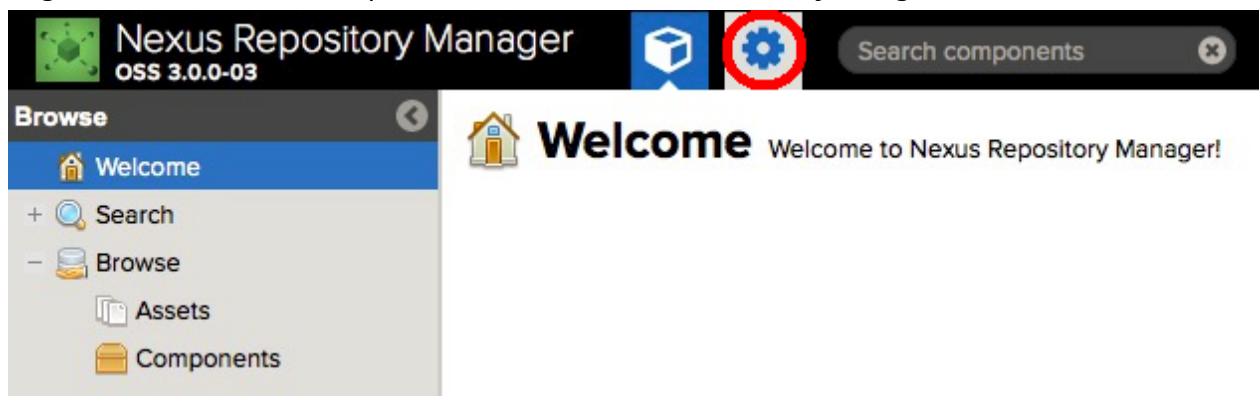
Si está ejecutando Docker en Linux, será la misma dirección IP que el host, que se puede encontrar usando el comando "ifconfig".

En mi caso, la dirección IP del host Docker es 192.168.1.72.

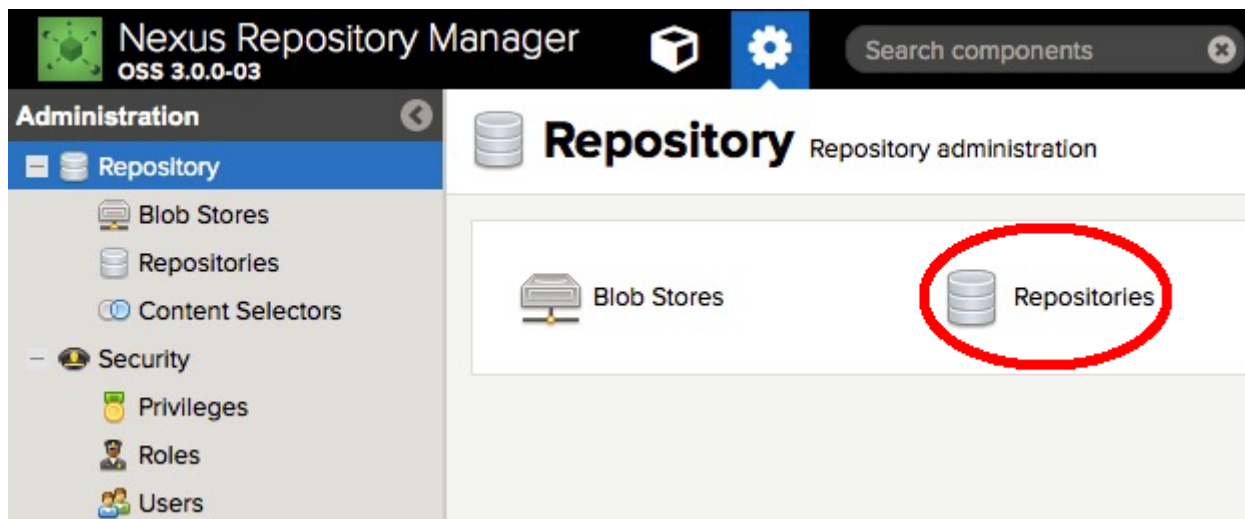
- Abra la URL [Dirección IP del host Docker]: 8081 en un navegador web.
- Inicie sesión en Nexus 3.

Haga clic en el botón Iniciar sesión en la esquina superior derecha y utilice el nombre de usuario "admin" y la contraseña "admin123". Si estas credenciales han cambiado, compruebe la [página de imagen de Nexus 3 Docker en DockerHub](#).

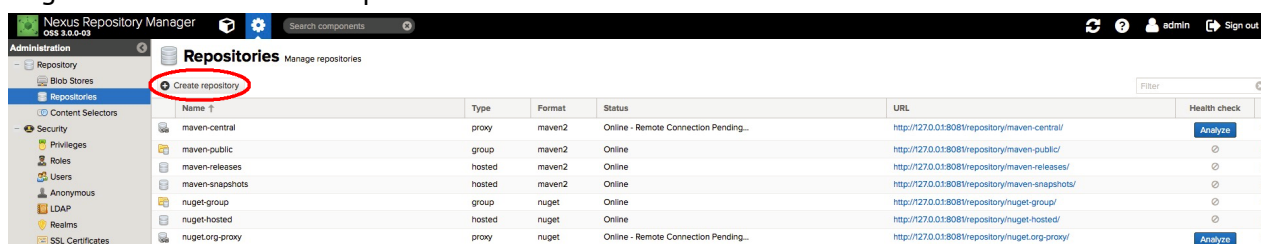
- Haga clic en la rueda dentada para ir a la sección de administración y configuración del servidor.



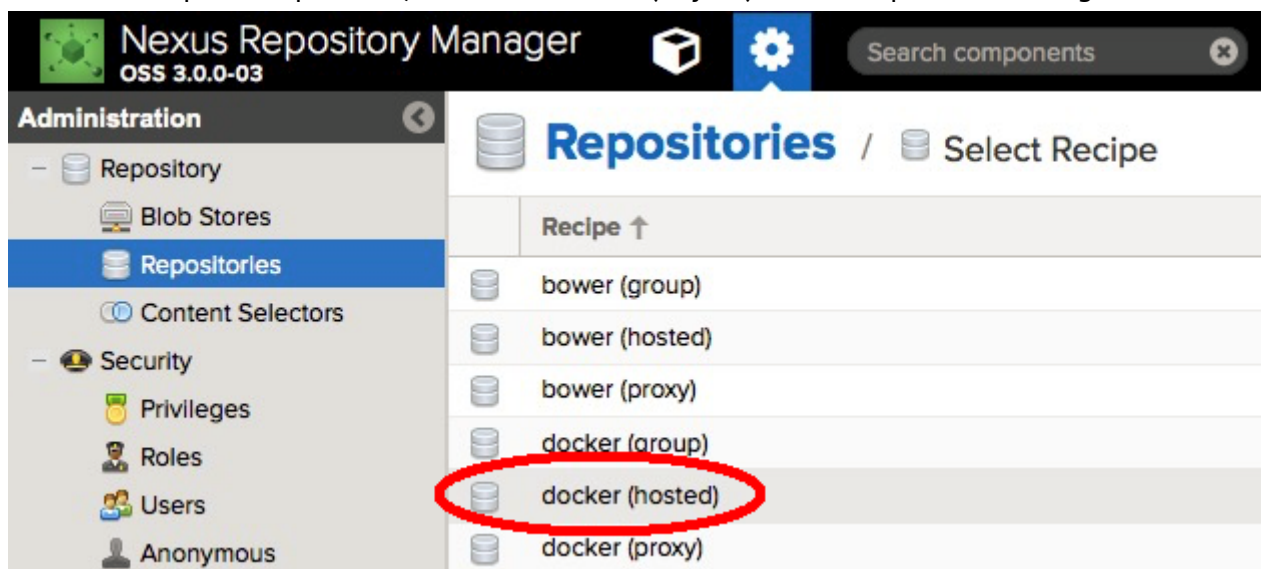
- Haga clic en Repositorios.



- Haga clic en el botón Crear repositorio.



- En la lista de tipos de repositorio, seleccione "docker (alojado)" como el tipo del nuevo registro.



- Configure el repositorio de Docker.

Dar al repositorio un nombre - en mi caso es "IvansDockerRepo".

Asegúrese de que la casilla de verificación en línea está marcada.

Marque la casilla de verificación HTTP en Conectores de repositorio e ingrese el número de puerto 8123.

Active la casilla de verificación Habilitar Docker V1 API.

Seleccione predeterminado en Blob store.

- Haga clic en el botón Crear repositorio.

Nexus Repository Manager OSS 3.0.0-03

Administration

- Repository
- Blob Stores
- Repositories
- Content Selectors
- Security
 - Privileges
 - Roles
 - Users
 - Anonymous
 - LDAP
 - Realms
 - SSL Certificates
- Support
 - Analytics
 - Logging
 - Log Viewer
 - Metrics
 - Support ZIP
 - System Information
- System
 - Bundles
 - Capabilities
 - Email Server
 - HTTP
 - Tasks

Repositories / Select Recipe / Create Repository: docker (hosted)

Name: A unique identifier for this repository
IvansDockerRepo

Online: ☒ If checked, the repository accepts incoming requests

Repository Connectors

Connectors allow Docker clients to connect directly to hosted registries, but are not always required. Consult our [documentation](#) for which connector is appropriate for your use case.

HTTP: ☒ Create an HTTP connector at specified port. Normally used if the server is behind a secure proxy.
8123

HTTPS: ☐ Create an HTTPS connector at specified port. Normally used if the server is configured for https.

Docker Registry API Support

Enable Docker V1 API:
☒ Allow clients to use the V1 API to interact with this Repository.

Storage

Blob store:
Blob store used to store asset contents
default

Strict Content Type Validation:
☒ Validate that all content uploaded to this repository is of a MIME type appropriate for the repository format

Hosted

Deployment policy:
Controls if deployments of and updates to artifacts are allowed
Allow redeploy

Create repository Cancel

El registro de Docker privado ahora está listo para ser utilizado.

Inicie sesión en el Registro de Docker Privado del Cliente

Antes de poder interactuar con el registro de Docker desde un cliente Docker, necesitamos iniciar sesión en el registro.

- Permitir conexiones inseguras Windows y OS X

Dado que hemos expuesto el registro de Docker privado en un punto final HTTP simple, necesitamos configurar el daemon de Docker que actuará como cliente (s) al registro de Docker privado para permitir conexiones no seguras. Esto puede no ser recomendable en todos los entornos, pero por el bien de este ejemplo, las conexiones no seguras son simples y bastarán.

- Busque el archivo config.json de Docker que contiene la clave "InsecureRegistry".

En Mac OS X y suponiendo que el nombre de la máquina Docker es "predeterminado", se encuentra en el directorio ~ /.docker / machine / machines / default.

En Windows, asumiendo de nuevo que el nombre de la máquina Docker es "predeterminado", se encuentra en el directorio C: \ Users \ [nombre de usuario] \ . Docker \ machine \ machines \ default.

La sección del archivo de configuración en la que se puede encontrar la clave "InsecureRegistry" tiene este aspecto:

```

1  "HostOptions": {
2    "Driver": "",
3    "Memory": 0,
4    "Disk": 0,
5    "EngineOptions": {
6      "ArbitraryFlags": [],
7      "Dns": null,
8      "GraphDir": "",
9      "Env": [],
10     "Ipv6": false,
11     "InsecureRegistry": [],
12     "Labels": [],
13     "LogLevel": "",
14     "StorageDriver": "",
15     "SelinuxEnabled": false,
16     "TlsVerify": true,
17     "RegistryMirror": [],
18     "InstallURL": "https://get.docker.com"
19   },

```

- Inserte el IP en el que se expone la interfaz gráfica de usuario de Nexus 3 junto con el número de puerto 8123 al valor de la clave "InsecureRegistry". En mi caso, el valor-clave modificado se verá así:

```
1 "InsecureRegistry": ["192.168.1.72:8123"],
```

- Si está utilizando Docker para Mac o Windows Beta:

Nuevo método (asegúrese de tener la última actualización): Los registros inseguros se pueden configurar en Preferencias de Docker, en la ficha Avanzadas.

~~Método antiguo: Cree un archivo llamado config.json con este contenido (reemplace la dirección IP según sea necesario):~~

~~{"Controlador de almacenamiento": "aufs", "depurar": true, "insecure-registries": ["192.168.1.72:8123"]}~~

~~El conjunto de la configuración mediante el comando:~~

~~Pinata set daemon @ config.json~~

- Reinicie la máquina Docker.

- Permitir conexiones no seguras Linux

Para permitir conexiones no seguras desde un cliente Docker que se ejecuta en Linux:

- Edite el archivo / etc / default / docker y agregue la siguiente línea:
DOCKER_OPTS = "\$ DOCKER_OPTS -insecure-registry = [Nexus3 Docker host IP]: 8123"
- R estart el servicio Docker utilizando: sudo service docker restart

- Log in al Registro de Docker Privado

Ahora debe ser capaz de iniciar sesión en el registro de Nexus 3 con el siguiente comando (reemplazar la dirección IP según sea necesario) usando el nombre de usuario "admin" y la contraseña "admin123", ambos sin comillas:

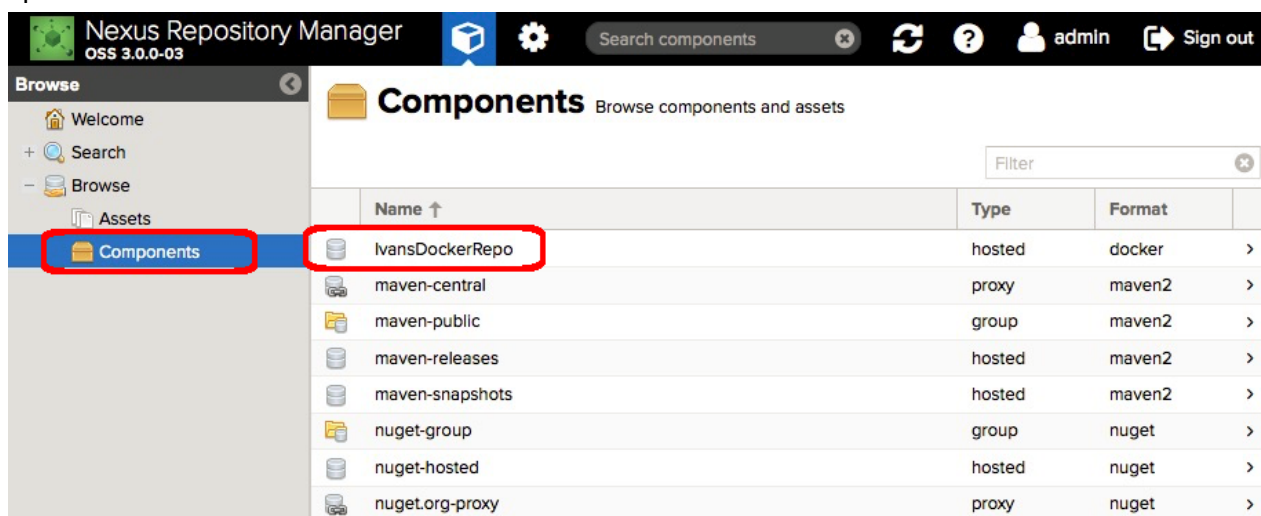
```
1 docker login -u admin -p admin123 192.168.1.72:8123
```

El resultado debe ser un mensaje diciendo que el inicio de sesión tuvo éxito.

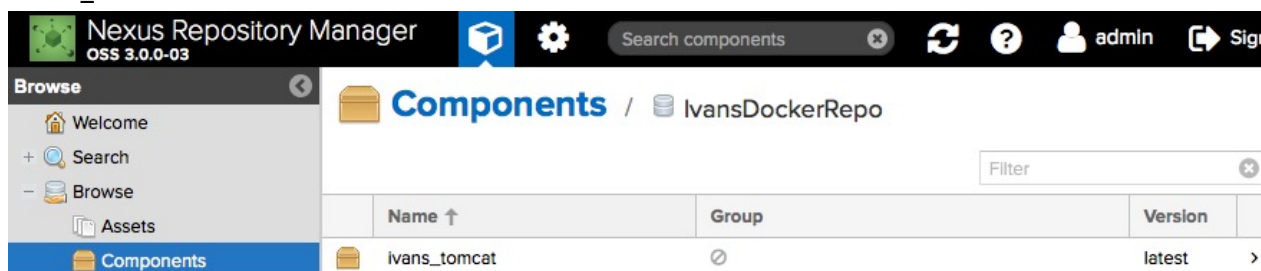
Empujar una imagen de Docker al registro privado

Después de haber sacado la imagen oficial de Tomcat antes y de haber iniciado sesión en nuestro registro privado, ahora estamos listos para empujar la imagen de Tomcat Docker a nuestro registro. En las instrucciones siguientes, reemplace la dirección IP del registro de Docker según sea necesario.

- Etiquete la imagen. Docker
tag tomcat: más reciente 192.168.1.72:8123/ivans_tomcat:latest
Esto le dice a Docker el nuevo nombre que se usará para la imagen de Docker y la dirección IP del registro a la que más tarde empujaremos la imagen.
- Empuje la imagen a nuestro registro privado.
Docker push 192.168.1.72:8123/ivans_tomcat:latest
- Abra la URL <http://192.168.1.72:8081> en un navegador.
- Haga clic en la categoría Componentes en la columna Examinar de la izquierda.
- Haga clic en el registro de Docker con el nombre "IvansDockerRepo" en la lista de componentes que aparecen.



- En la lista de componentes en IvansDockerRepo, debe haber una sola entrada con el nombre "ivans_tomcat".



- Para extraer la imagen, tal vez en otro equipo, utilice los siguientes comandos.

```
1 docker login -u admin -p admin123 192.168.1.72:8123
2 docker pull 192.168.1.72:8123/ivans_tomcat:latest
```

Esto concluye este artículo.

Codificación feliz!

Categoría: Desarrollo de Software Etiquetas: docker , docker registry , nexus , nexus 3

6 pensamientos en " Crear un registro de Docker privado "



Amir

17 de enero de 2017

Gracias por el gran artículo. Me pregunto por qué no utilizó oficial docker registro (ahora lo llaman distribución) en lugar nexus? ¿Hay algún beneficio en el uso de nexus cuando sólo necesitamos el registro docker y no otras bases de código proporcionado por nexus (por ejemplo, bower, npm, etc)?



Ivan Krizsan

Autor del mensaje

17 de enero de 2017

The reason for using Nexus is, which is not in the article, that I have been using Nexus as a Maven repository and in this way I can use one and the same repository management solution.

I haven't really tried the official Docker registry so I cannot compare it to Nexus.

Happy coding!



Andre

January 24, 2017

Next best thing about using Nexus: You can configure the scheduler to free up space from deleted/orphaned images and layers.. Something that the official docker registry cannot do right now.

On the downside: I couldn't easily configure Nexus as a caching proxy(which should work in theory, but fails at Nexus' inability to allow anonymous access to the proxy and Docker's failure to re-prompt authentication when hitting a 401)...

Also the API of Nexus 3 leaves quite a few loose ends.. If you're ok with clicking your configuration, it's fine.. If you try to make it work with the scripting API, it's not 😊



tolbin
May 7, 2017

i have try your tutorial but i have a little problem with "docker login -u admin -p admin123 my_ip:8123"
the respons like this 😞 "Error response from daemon: Get https://my_ip:8123/v1/users/: http: server gave
HTTP response to HTTPS client
"



Tuelho
June 28, 2017

Try docker login -u admin -p <http://admin123> your_ip:8123



Tuelho
June 28, 2017

Thanks!! Works fine to me! But I needed to use http://my_ip:8123 to login.

Iconic One Theme | Desarrollado por Wordpress