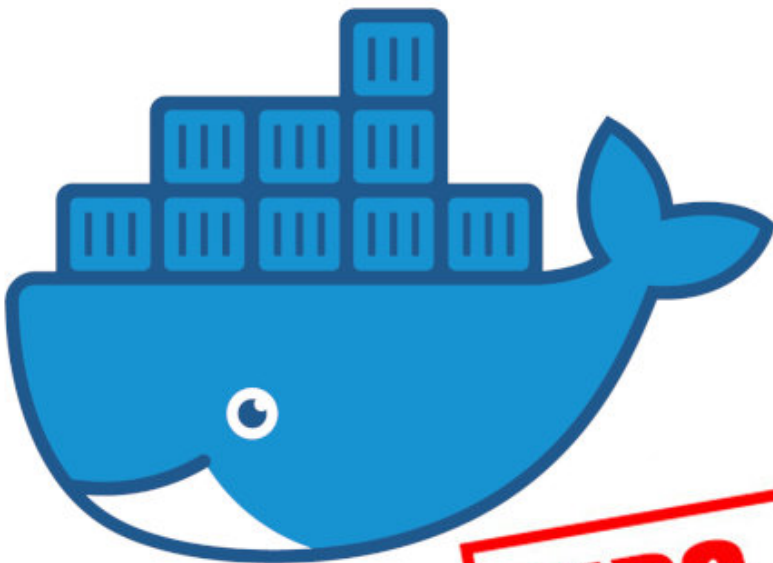


Artículo

[Ricardo Paiva](#) · Mar 11, 2021 Lectura de 6 min

## Cómo gestionar el espacio en disco en Docker

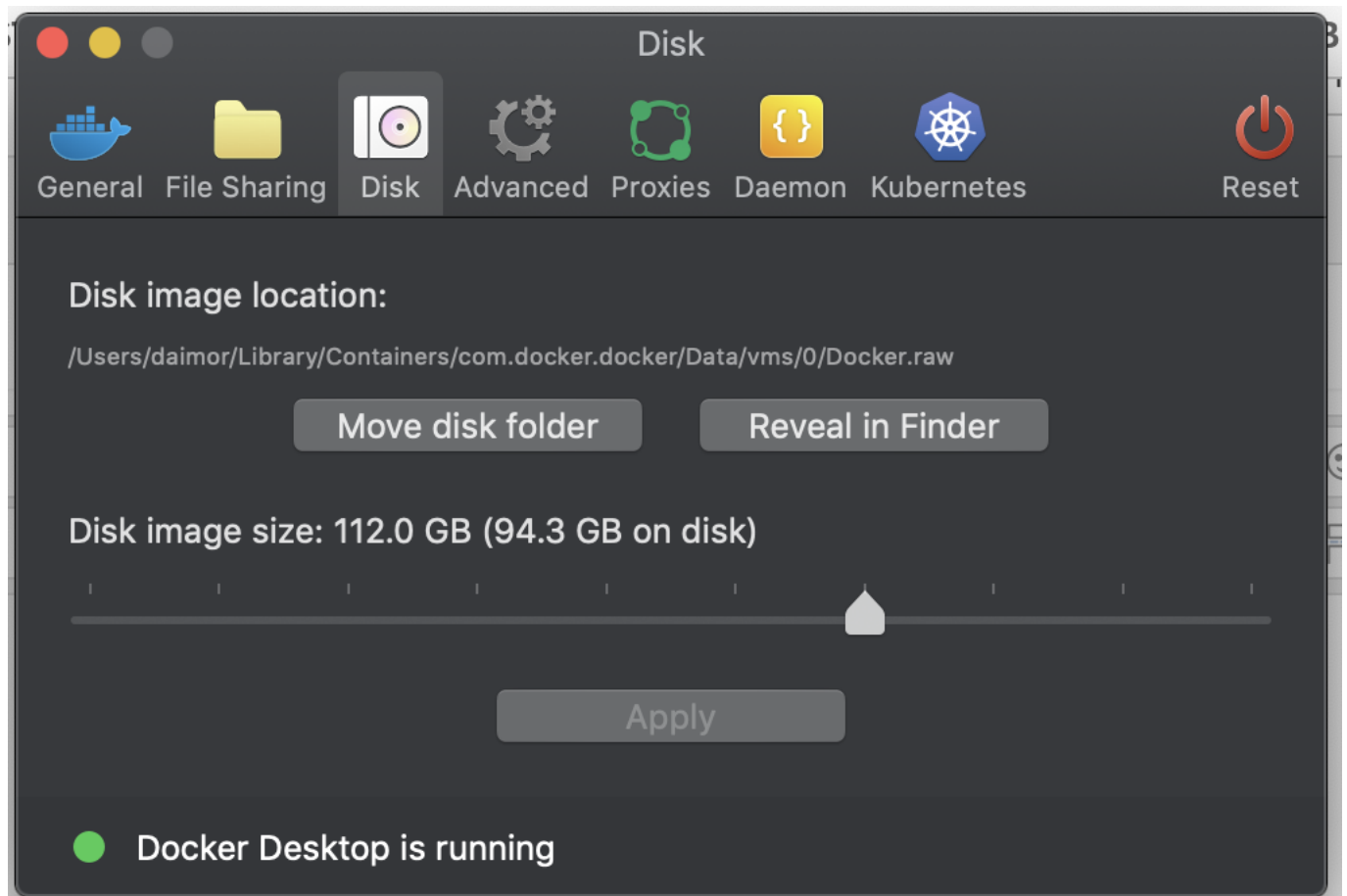
Quería escribirlo como comentario al [artículo](#) de [@Evgeny Shvarov](#). Pero resultó demasiado largo, así que decidí publicarlo por separado.



**TIPS & TRICKS**

Me gustaría añadir una pequeña aclaración sobre cómo utiliza Docker el espacio en disco y como limpiarlo. Yo uso macOS, por lo tanto todo lo que explico aplica principalmente a macOS, pero los comandos de Docker se adaptan a cualquier plataforma.

Cuando Docker se incluye en Linux, por defecto funciona en el mismo sistema de archivos. Pero en Windows y macOS, funciona en una pequeña máquina virtual con su propio Linux dentro. Y el espacio del disco está limitado por mi configuración en Docker. En mi caso, lo configuré para utilizar hasta 112 GB.



Por lo tanto, cuando trabajes de forma activa con Docker, tu espacio interior dejará de usarse. Puedes comprobar como Docker emplea todo ese espacio con el comando:

```
$ docker system df
```

TYPE	TOTAL	ACTIVE	SIZE	RECLA
IMABLE				
Images	84	6	66.02GB	55.6G
B (84%)				
Containers	6	5	4.914GB	0B (0
%)				
Local Volumes	19	4	1.812GB	342.7
MB (18%)				
Build Cache	0	0	0B	0B

En macOS con las últimas versiones de Docker, se utiliza el formato en bruto del disco (anteriormente era qcow2). Y junto con el sistema de archivos APFS en macOS, este archivo puede ocupar menos espacio físico que el propio tamaño del archivo. Observa estos dos comandos.

```
$ ls -lh ~/Library/Containers/com.docker.docker/Data/vms/0/Docker.raw
-rw-r--r--@ 1 daimor  staff   104G Jul 13 15:49 /Users/daimor/Library/Containers/com.
docker.docker/Data/vms/0/Docker.raw
```

```
$ du -h ~/Library/Containers/com.docker.docker/Data/vms/0/Docker.raw
88G    /Users/daimor/Library/Containers/com.docker.docker/Data/vms/0/Docker.raw
```

El comando ls muestra el tamaño de mi archivo Docker.raw como 104Gb, mientras que el comando du muestra el tamaño real en el disco, que es de 88Gb.

Bien, docker system df me mostró que puedo recuperar algo de espacio. Vamos a hacerlo.

```
$ docker system prune -f
Deleted Containers:
79b3d54ae5a881e37771cfdcd1d651db9ce036abc297dc55bdd454eb287f0e329

Deleted Images:
deleted: sha256:298d555976effb112428ed3f6bcc2f4d77ab02b4f287a230d9535001184078f5
deleted: sha256:adb2c64ce6e44d837fce8067c7498574822bff90ed599d1671c126539fe652ac
deleted: sha256:9695172139ceec16f1071449daf29bd1c424353044088b92b8acbf33f59952e67
deleted: sha256:24d834b252e25e645b8b5d9194360f5ab1a26ffd2b5c03b6593b9a2c468f59fa
deleted: sha256:1b4e3e73fe0b7d88d5ec718bdc6dc6d17d9fe8ba00988eb72690d76f2da3d1a3
deleted: sha256:9f218f6c7aca9c21760ae43590a2d73b35110e10b6575125ed3ccd12c4495d6e
deleted: sha256:b2fa3335d672a0dc60ea7674c45ee3c85b9fc86584a0e21cc7f1900c368ceec3
deleted: sha256:2ecace396ab65fd393dfb2e330bece974cd952e7a41364352f9c867d9ea4c34e
deleted: sha256:16b894351fe53b95dd43d7437bbcd5104b8613bc1fa8480826e843d65fc92a3
deleted: sha256:b00d9c05035eac62f3ed99a814cd6feea3e4b68797b6d1203e2f41538c78c2aa
deleted: sha256:5a3d0d9f36b356cb47d3838585da6450c60e2860ef143d1406b48d3a5e72b92b
deleted: sha256:998e719368ff74d13b3a8c096ce81f8f2c4bb28bd1ccd169bfa173b4a78d2e74
deleted: sha256:a74d7ff2ca7d623134f9ce1db40da476134a733935a3f322ba34b99653c6273d
deleted: sha256:4d0dcd2bdad2cf0cb91d13313afff29326771bdac27fcb8780545687dbd39ae4
deleted: sha256:29a8989eed3d4002053f98bf562654910ee5f8836940daaa2f2344a8f29a52a2
deleted: sha256:12d34fbf938d19b193199ea6cce5d690fd0d57ec3f4d1630e1d4b3790379c9ec
deleted: sha256:75aba481bb5ccaa52a3aadf311ae22485fb2a82d69be864fe2f45f2834c5e515
deleted: sha256:326efafee9b92e06876878b21a2931ba771bc0e0b2b359f906ef6cca1d297905
deleted: sha256:913937f4ea932fcb00b6c6b3007970296955aa4f488d6fbaa1a575a5aa4ff5ab
deleted: sha256:f3fc0c75858a36ff9d3f4e8eb7a96f511158bbac92d128760b0d3340d828c5da
deleted: sha256:c002dde1ea6a02ae3e3037442a5c556a925e3e4750a6b2aa923c51fa3d11f5ac
deleted: sha256:e763f6e226613c67aaf5984e4c74b9f6e9e28e0490a4f3286885f498a57d3fa0
deleted: sha256:e7daf0a1574376c8602515dc70e44392d16e1b79013d6e81a9b697794432e660
deleted: sha256:ce33670f78109dcacc73a7c6d70f4b5cd4a13bcfe7878b9df5e4e16b812e5df4
deleted: sha256:95bf79e86f83ed16943f9c34035bf8167a4b89466a05d6793c2957d6d46bab2d
deleted: sha256:056d184391613b33303ccf3818a95a17398e9da813d093a6ee5d87952539380c
```

Total reclaimed space: 5.537GB

Este comando elimina cualquier contenedor detenido y cualquier imagen no etiquetada que no se esté utilizando por cualquier imagen etiquetada. Y se puede eliminar de forma segura.

Tal vez has notado que solo se recuperaron 5.5 GB, mientras que docker system df hablaba de unos 55 GB. Eso es porque df cuenta todas las imágenes no activas, no solo las activas. Si también quieres eliminar todas esas imágenes, puedes utilizar este comando, lo que elimina cualquier imagen que no se utilice en los contenedores que estén ejecutándose en este momento. Por lo tanto, si no tienes ningún contenedor funcionando, eliminará todas las imágenes locales.

```
docker system prune -a
```

Acabo de recuperar solo las imágenes activas y los contenedores detenidos. Cuánto espacio utiliza mi docker ahora.

```
$ docker system df
```

TYPE	TOTAL	ACTIVE	SIZE	RECLA
IMABLE				
Images	83	5	60.48GB	50.1G
B (82%)				
Containers	5	5	4.914GB	0B (0

```
% )
Local Volumes      19              3              1.812GB        342.7
MB (18%)
Build Cache        0              0              0B             0B
```

Como puedes ver, ya utiliza menos tamaño. Is mostrará el mismo resultado. El tamaño del archivo principalmente crece.

```
$ ls -lh ~/Library/Containers/com.docker.docker/Data/vms/0/Docker.raw
-rw-r--r--@ 1 daimor  staff   104G Jul 13 16:07 /Users/daimor/Library/Containers/com.
docker.docker/Data/vms/0/Docker.raw
```

Pero para macOS es más importante cuánto espacio se utiliza en un disco físico.

```
$ du -h ~/Library/Containers/com.docker.docker/Data/vms/0/Docker.raw
69G    /Users/daimor/Library/Containers/com.docker.docker/Data/vms/0/Docker.raw
```

Y como puedes ver ahora son 69 GB, que son aproximadamente 19 GB menos de los que eran anteriormente.

Así que, para los usuarios de macOS, realmente no importa el tamaño del archivo, con las optimizaciones de APFS en realidad puede ser menor.

Otra forma es reducir las imágenes antiguas con algún filtro por fecha de creación. Al igual que este ejemplo, se eliminarán todas las imágenes creadas hace más de 10 días, pero se mantendrán las imágenes que actualmente utilizan los contenedores.

```
$ docker image prune --all --filter until=240h
```

[#Administración del sistema](#) [#Contenedorización](#) [#Docker](#) [#Nube](#) [#Caché](#) [#InterSystems IRIS](#)

URL de fuente: <https://es.community.intersystems.com/post/c%C3%B3mo-gestionar-el-espacio-en-disco-en-docker>