

The 2025 Developer Survey results are in. Explore insights into technology and tools, careers, community and more. [View results.](#)



how to clean up docker overlay directory?

Asked 10 years ago Modified 1 year, 5 months ago Viewed 227k times Part of AWS Collective



I'm running docker via CoreOS and AWS's ECS. I had a failing image that got restarted many times, and the containers are still around- they filled my drive partition. Specifically,

126

`/var/lib/docker/overlay/` contains a large number of files/directories.



I know that [docker-cleanup-volumes](#) is a thing, but it cleans the `/volumes` directory, not the `/overlay` directory.



`docker ps -a` shows over 250 start attempts on my bad docker container. They aren't running, though.

Aside from `rm -rf /var/lib/docker/overlay/*`, how can I/should I clean this up?



amazon-web-services

docker

coreos

Share Improve this question Follow

asked Jul 29, 2015 at 22:29



[300D7309EF17](#)

24.8k ● 14 ● 90 ● 103

12 Answers

Sorted by: Highest score (default)



From our side we used:

223

```
sudo docker system prune -a -f
```



Which saved me 3Go!



We also used the famous commands:



```
sudo docker rm -v $(sudo docker ps -a -q -f status=exited)
sudo docker rmi -f $(sudo docker images -f "dangling=true" -q)
docker volume ls -qf dangling=true | xargs -r docker volume rm
```

We put that on cron to manage a little bit more efficiently our disk space.

Reference: <https://forums.docker.com/t/some-way-to-clean-up-identify-contents-of-var-lib-docker-overlay/30604/4>

Disclaimer

Careful with that 3rd command (... xargs -r docker volume rm). Make sure the containers that may use those volumes are running when you run this. Otherwise, they will be seen as dangling, and therefore deleted.

Share Improve this answer Follow

edited Feb 19, 2024 at 20:13

answered Sep 11, 2017 at 15:51



Robert Sinclair

5,486 ● 2 ● 49 ● 52



user3774109

2,496 ● 1 ● 14 ● 13

- 5 Life saver! I had no idea why `df` said my ec2 volume was full until I ran `ncdu` on my `/var` directory which led me here! I reclaimed 280+ GB from that first command! – Josh Yolles Sep 22, 2020 at 18:22
- 4 Careful with that 3rd command (... xargs -r docker volume rm). Make sure the containers that may use those volumes are running when you run this. Otherwise, they will be seen as dangling, and therefore deleted. – Julien Oct 7, 2020 at 23:29
- 1 There is a lot of discussion on this: github.com/moby/moby/issues/33775 A lot of people have run into the overlay directory consuming more space than expected and that it isn't clear how to clean it up. – Mnebuerquo Apr 22, 2022 at 20:33
- 1 Please pay attention, the command `docker system prune -a -f` will remove any stopped container. If you have a container stopped for a moment and you plan to use it in the future, do not use `-a` option. – Dorin Jan 18, 2024 at 13:48



docker builder prune

21

this helps cleaning up the docker overlay2 folder



Share Improve this answer Follow

answered Apr 27, 2023 at 6:54



tomitomsen3000

211 ● 2 ● 3



- 1 Just cleared 50GB with this. Apparently the build layers are also cached in here, so its not only used by running containers. – AFP_555 Oct 4, 2023 at 15:35
- 1 THIS!! Thank you. I cleared 330GB with this. My overlay2 directory had grown to 400+GB. I always do a image or system prune, but I was never able to really clean up the hundreds of GB. In the past I would shutdown docker service, rm -rf the overlay directory, then restart docker. But now I don't have to lose everything to clean up space. – Cristhian Ulloa Oct 17, 2024 at 14:02



Here's the hacky way I'm doing this right now. I'm not going to accept it as an answer because I'm hoping there's a better way.

15

```
# delete old docker processes
docker rm `docker ps -a | grep Exited | awk '{print $1 }'`
  ignore_errors: true
```



```
# delete old images. will complain about still-in-use images.
docker rmi `docker images -aq`
```



Share Improve this answer Follow

answered Aug 8, 2015 at 3:26



[300D7309EF17](#)

24.8k ● 14 ● 90 ● 103

2 `awk` can filter: `docker ps -a | awk '/Exited/{ print $1 }'`. Nice hack though.

– [Krzysztof Jabłoński](#) Feb 26, 2018 at 15:42



I have added this to `bashrc` in my dev environment, and gotten used to running it every day or so.

14

```
function cleanup_docker() {
  docker ps -f status=exited -q | xargs -r docker rm
  docker images -f dangling=true -q | xargs -r docker rmi
}
```



In some cases, the following script can free up more space, as it will try to remove all images, and just fail silently:



```
function cleanup_docker_aggressive() {
  for i in $(docker images --no-trunc -q | sort -u)
  do
    docker rmi $i 2> /dev/null
  done
}
```

Sadly, they're not significantly cleaner than your solution.

EDIT: Starting with Docker 1.13, you can use [docker system](#):

```
docker system df # to check what is using space
docker system prune # cleans up also networks, build cache, etc
```

EDIT: Starting with Docker 2017.09, you can also use [container](#) and [image](#)

```
docker container prune
docker image prune -a
```

the latter you can use with fancy filters like `--filter "until=24h"`

Share Improve this answer Follow

edited Nov 20, 2018 at 7:56

answered Sep 14, 2015 at 12:24



analytik

792 ● 7 ● 16

1 I've cleaned everything but overlay directory is still 20GB+ while active containers+images use only 6GB together – [vladkras](#) Sep 4, 2017 at 13:04

1 @vladkras - If you can, try restarting the docker containers one by one (not just halt+resume) and then run the cleanup again. Likely there's something producing a lot of logs. If that doesn't help, maybe you're not counting the supposed usage correctly? Also, I highly recommend switching to overlay2, unless some service is producing a ton of log output, you should save 60%+ of space. – [analytik](#) Sep 4, 2017 at 16:34

save me a half of my sda, `docker version`

```
Client: Version: 18.09.6 API version: 1.39 Go version: go1.10.8
```

– [Tiana987642](#) Mar 1, 2023 at 20:04

It's not real!

11

`/var/lib/docker/overlay2`, as the name suggests, contains a bunch of `overlay` file systems (good intro [here](#)). Run `mount | grep overlay2` and you will see that all the

`/var/lib/docker/overlay2/*/merged` folders are of mounts of type `overlay`. This means that *what `du` says isn't real*. To get a sense of how much space is *actually* being used in those folders (on disk) you need to limit your attention to just the `upper` directory of the overlays mount (called `diff` in docker's case), e.g.:

```
du -sch /var/lib/docker/overlay2/*/diff
...
3.8G total
```

for comparison, in my case:

```
> du -sch /var/lib/docker/overlay2
...
17G total
```

Update

It seems that you can also simply use `du 's -x / --one-file-system` option ("skip directories on different file systems") to only see the real part:

```
> du -schx /var/lib/docker/overlay2
```



Christian Fritz

21.5k ● 3 ● 48 ● 82

So if 'du' say all space is used on the disk, and it's being used by the overlay, how does saying its not real help? My disk writes are now failing...so it's real enough to interfere with operations. Is the a command to align real with reported disk usage? – TSG Oct 17, 2023 at 20:56

`du` doesn't tell you anything about the disk being full or not. Do you perhaps mean `df` ? If so, what `df` says is always real. And I believe your question is answered by my answer, which tells you how to only see the real part. – Christian Fritz Oct 18, 2023 at 0:17

I meant 'df' sorry – TSG Oct 18, 2023 at 1:29



We just started having this problem, and btafarelo's answer got me part of the way, or at least made me feel better about removing the sha256 entries.

10

System info: ec2 instances running CoreOS 1.12 behind an ELB



- Drain the docker instance from the ELB
- Shutdown docker



```
systemctl stop docker
rm -rf /var/lib/docker/overlay/*
```

- Execute the results of the commands

```
for d in $(find /var/lib/docker/image/overlay -type d -name '*sha256*'); do echo
rm -rf $d/* ; done
```

- reboot (easiest way to bring everything back up)

This recovered about 25% of the disk after the services restarted with no ill side affects.



Kevin McFadden

139 ● 1 ● 4

tried your cleanup and ended having all service creation rejected after with error message `lstat /var/lib/docker/overlay` – tkyass Jun 15, 2017 at 22:14

- 1 did you remove the overlay directory by accident? `lstat` is for symlinks, but mine isn't a symlink. – Kevin McFadden Jun 17, 2017 at 1:59 ✎

Even after stopping docker I still had some locked folders in `/var/lib/docker/overlay/`. I had to manually `umount` them so that I could delete them. – Joel B Jun 26, 2017 at 1:57



docker ps

7



- --quiet
- --all
- --filter status=exited



docker rm

- --force

docker images

- --quiet
- --all
- --filter dangling=true

docker rmi

- --force

Your hacky way is fine.

```
docker rm `docker ps -a | grep Exited | awk '{print $1 }'`
```

My hacky way is

```
docker rm $(docker ps --all | awk '/ago/{print $1}')
```

A slightly cleaner way is to run `docker ps` with the `--quiet` (-q) flag to get just the id number and `--filter status=exited` to --filter just the exited ones.

```
docker rm $(docker ps --filter status=exited --quiet) # remove stopped docker processes
```

or to run `docker rm` with the `--force` (-f) flag and `docker ps` with the `--all` (-a) flag to shut down even the running ones

```
docker rm --force $(docker ps --all --quiet) # remove all docker processes
```

What's probably taking up all that disk space after several failed builds is the images. To conserve disk space on the docker host, periodically remove unused docker images with

```
docker rmi $(docker images --filter dangling=true --quiet) # clean dangling docker images
```

or to get more aggressive, you can `--force` (-f) it to clean up `--all` (-a) images

```
docker rmi --force $(docker images --all --quiet) # clean all possible docker images
```

@analytik 's way of putting it into a [.bashrc](#) function seems like a practical idea

```
function cleanup_docker() {
  docker rm --force $(docker ps --all --quiet) # remove all docker processes
  docker rmi $(docker images --filter dangling=true --quiet) # clean dangling docker images
}
```

and if you're in the habit of generating lots of docker images that you don't need, add it to [.bash_logout](#)

Share Improve this answer Follow

answered Dec 4, 2015 at 18:03



woodvi

2,026 ● 24 ● 27



3



Please be aware that `docker prune` commands do not clean `/var/lib/docker/overlay2` directory. It is also not advised to remove only the overlay directory as it may impact existing containers. I have searched for a lot of articles but couldn't find any solution to clean overlay directory other than cleaning the entire docker state:

```
# Please understand that this will restart the docker engine in a completely empty state
# i.e. you will lose all images, containers, volumes, networks, swarm state, etc.
# You can obviously first take the backup of the directories that you want to keep and copy the contents back after restarting docker service.
```

```
service stop docker
rm -rf /var/lib/docker
service start docker
```

The last command brings the docker service back up with all the folders inside `/var/lib/docker` again.

Share Improve this answer Follow

answered Apr 30, 2022 at 7:24



Rahul Kumar
849 ● 8 ● 8



I followed these simple steps

2

Step 1: `df -h` [checked the memory used, to be sure, memory is used by overlay folder].



Step 2: `sudo docker system prune` [this cmd removes all unused containers/images/networks]



Step 3: `sudo docker image prune -a` [for any dangling images, if present]



Step 4: `df -h` [to be sure, overlay data is removed].

Share Improve this answer Follow

answered May 21, 2020 at 14:55



Aniket J kamble
140 ● 1 ● 8



Docker garbage collection can be done in an easy way using another docker container

<https://github.com/spotify/docker-gc>

1

You could make it run as a cron using <https://github.com/flaccid/docker-docker-gc-crond>



Share Improve this answer Follow

answered Oct 1, 2016 at 7:04



Forever
7,537 ● 10 ● 56 ● 59



Here is a working option:

1

```
docker rm -f $(docker ps -a |awk 'NR>1&&/Exited/{print $1}')
```



Share Improve this answer Follow

answered Feb 21, 2020 at 18:14



Dudi Boy
1



here is resolution to clean docker overlay directory from <https://lebkowski.name/docker-volumes/>

0

```
docker images --no-trunc | grep '<none>' | awk '{ print $3 }' | xargs -r docker rmi
docker ps --filter status=dead --filter status=exited -aq | xargs docker rm -v
```



for Docker < 1.9 :



```
find '/var/lib/docker/volumes/' -mindepth 1 -maxdepth 1 -type d | grep -vFf <(docker ps -aq | xargs docker inspect | jq -r '[]|.Mounts|.[]|.Name|select(.)')
```

Or for Docker >=1.9 :

```
docker volume ls -qf dangling=true | xargs -r docker volume rm
```

Share Improve this answer Follow

answered May 9, 2018 at 14:01



Wolverine

49 ● 4

Start asking to get answers

Find the answer to your question by asking.

Ask question

Explore related questions

amazon-web-services docker coreos

See similar questions with these tags.