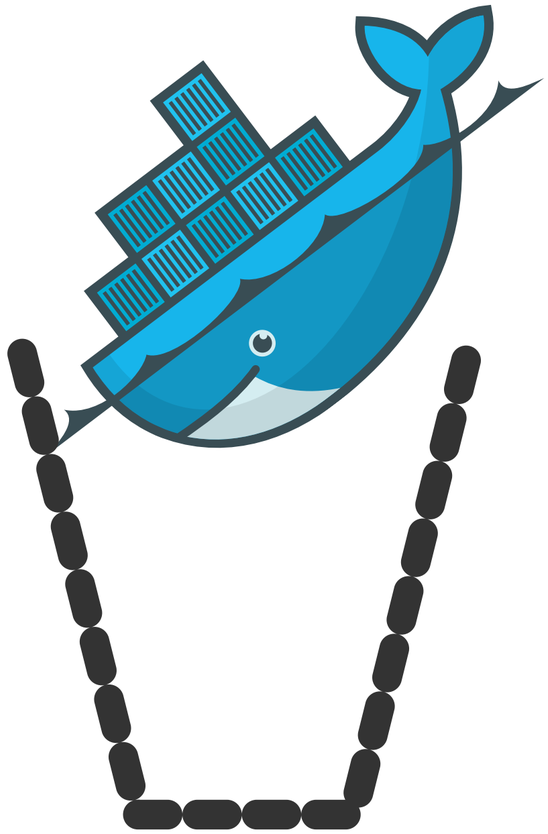
JULY 7, 2023

Docker

GARBAGE COLLECTION



Why is Docker cleanup important?

- If you use Docker for development, you probably run multiple containers, creating new images and giving a lot of space on your disk to things you don’t need.

- If you don’t address the storage issue at the right time, your root directory will quickly fill up the available disk space.

Why does Docker not do auto-cleanup?

- Docker takes a conservative approach to cleaning up unused objects (often referred to as “garbage collection”), such as images, containers, volumes, and networks: these objects are generally not removed unless you explicitly ask Docker to do so.

Show Docker disk usage

- docker system df

```
(base) Nadines-MacBook-Pro:project_code nadine$ docker system df
TYPE            TOTAL     ACTIVE    SIZE      RECLAIMABLE
Images          3         0         1.455GB   1.455GB (100%)
Containers      0         0         0B        0B
Local Volumes   206       0         25.74GB   25.74GB (100%)
Build Cache     24        0         510B      510B
```

Show detailed information on space usage

- docker system df -v

```
(base) Nadines-MacBook-Pro:project_code nadine$ docker system df -v
Images space usage:

REPOSITORY                      TAG       IMAGE ID       CREATED        SIZE      SHARED SIZE   UNIQUE SIZE   CONTAINERS
project_code-etl_job            latest    6308c1f630d1   2 hours ago    1.21GB    996.6MB       216.9MB       0
project_code-slackbot           latest    4a1633c06c30   2 hours ago    1.2GB     996.6MB       205.3MB       0
project_code-reddit_collector   latest    9c04effe10b1   2 hours ago    1.03GB    996.6MB       35.89MB       0

Containers space usage:

CONTAINER ID   IMAGE     COMMAND   LOCAL VOLUMES   SIZE      CREATED   STATUS    NAMES

Local Volumes space usage:

VOLUME NAME                                                        LINKS     SIZE
597535fa09b438a6088475451ca84bdaa2e7c166bd0da84977ce217677b28106   0         47.38MB
4f42b36c7d042a5eea001bf3288c343347dc27db31a21303cc4d33cdbcd03aa7   0         314.8MB
7fcf11e775d465afa9f590131c8829898c3c2f620bfa70b270d97ce8a07a8d42   0         47.3MB
af38d099ba4743435e5c1c1fdce8cdb669ce5e016a56e3c198e349b77a588346   0         47.3MB
b9faedbfa8ad6c9fb5a78c6f41bfac74d1102de1919a03368236846210e03470   0         47.3MB

CACHE ID       CACHE TYPE     SIZE      CREATED       LAST USED     USAGE     SHARED
74ri3wfxuwtm   regular        0B        2 hours ago   2 hours ago   1         true
qzshu6hx0hzt   regular        0B        2 hours ago   2 hours ago   1         true
h615sflbl1sy   regular        0B        2 hours ago   2 hours ago   1         true
r97tcrkvi12v   regular        0B        2 hours ago   2 hours ago   1         true
```

Definition

- Local Volumes
  - refers to Docker volumes that are created and managed locally on the Docker host machine. These volumes are used to persist and store data generated or consumed by containers.
  - Docker local volumes are separate entities from container filesystems. They provide a way to store and manage data outside the lifecycle of a container.
  - Local volumes can contain various types of data:
    - configuration files, logs, databases, or any other files generated or consumed by your application running inside a container

Why are volumes not removed with containers and images?

- When you remove a container or an image, Docker does not automatically delete associated volumes by default. This behavior is intentional to **prevent accidental data loss**.
- Docker assumes that volumes may contain important or user-generated data that should be preserved even if containers or images are removed.

Definition

- Build Cache
  - refers to the cache of intermediate build artifacts that are generated during the Docker image build process. It is used to optimize subsequent builds by reusing previously built layers and reducing the time and resources required for building Docker images.

  - Keep in mind that removing the build cache may increase the build time for subsequent image builds, as Docker will need to rebuild all layers from scratch.

How to cleanup space in Docker

- **Image**
  - Delete an image
    - *docker rmi <image>*
  - Delete dangling images
    - *docker image prune*
  - Delete all unused images
    - *docker image prune -a*

How to cleanup space in Docker

- **Containers**
  - Delete a container
    - *docker rm <container>*
  - Delete a running container
    - *docker rm -f <container>*
  - Delete stopped containers
    - *docker container prune*

How to cleanup space in Docker

- **Volume**
  - List the volumes
    - *docker volume ls*
  - Delete a volume
    - *docker volume rm <volume>*
  - Delete volumes
    - *docker volume prune*

Prune Everything

- docker system prune -a
  - prunes images, containers, and networks. **Volumes** are not pruned by default.

- docker system prune -a --volumes
  - prunes images, containers, volumes and networks not being used.

If cleaning did not immediately free up space:

- **File system overhead:** The reported size of the local volume might not precisely reflect the actual disk space occupied.
- **Delayed space reclamation:** This delay occurs because the file system marks the space as available for reuse but might not immediately reclaim it.
- **Residual data:** The local volume might have contained other files or directories apart from the data you explicitly removed. It's possible that some files or directories within the volume were not deleted, either due to permissions issues or incomplete cleaning. In such cases, the residual data would continue to occupy disk space.

Before

Macintosh HD - 3.23 GB available of 121.02 GB

After

Macintosh HD - 16.86 GB available of 121.02 GB

References

https://middleware.io/blog/docker-cleanup/
https://docs.docker.com/
ChatGPT