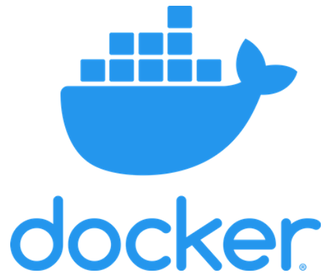
Docker Deep Dive

Chapter #5



Docker Images

Docker

IMAGES

DOCKER IMAGE

Docker image is like a stopped container.

MULTIPLE LAYERS

Images are made of multiple layers stacked on top of each other.

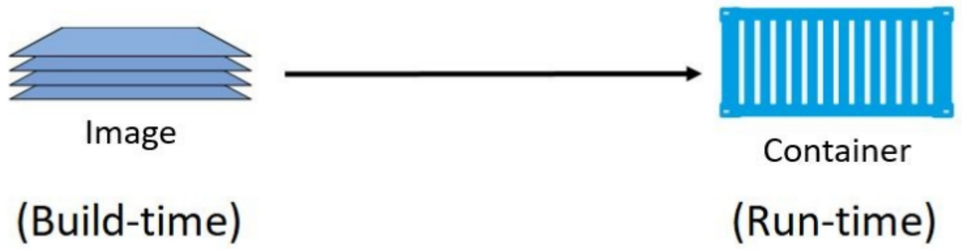
DOCKER HUB

Most popular image registry.

Docker Image

Docker Images: deep dive

Images are like stopped containers (or classes if you're a developer). In fact, you can stop a container and create a new image from it. With this in mind, images are considered build-time constructs whereas containers are run-time constructs.



Images & containers

Once we've started a container from an image, the two constructs become dependent on each other and you cannot delete the image until the last container using it has been stopped and destroyed.

Attempting to delete an image without stopping and destroying all containers using it will result in error.

```
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
deepone             latest              0d34f8b9483d        3 hours ago         279MB
alpine              latest              a24bb4013296        4 days ago          5.57MB
centos              latest              470671670cac        4 months ago        237MB
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
S
65509b0b4609        deepone:latest      "/usr/sbin/httpd -D …"   About an hour ago   Up About
x
[root@ip-172-31-25-230 ~]# docker image rm deepone:latest
Error response from daemon: conflict: unable to remove repository reference "deepone:latest" (
s using its referenced image 0d34f8b9483d
[root@ip-172-31-25-230 ~]#
```

Images are small

The whole purpose of a container is to run an application or service. This means that the image a container is created from must contain all OS and application files required to run the app/service. However, containers are all about being fast and lightweight. This means that the images they're built from are usually small and stripped of all non-essential parts.

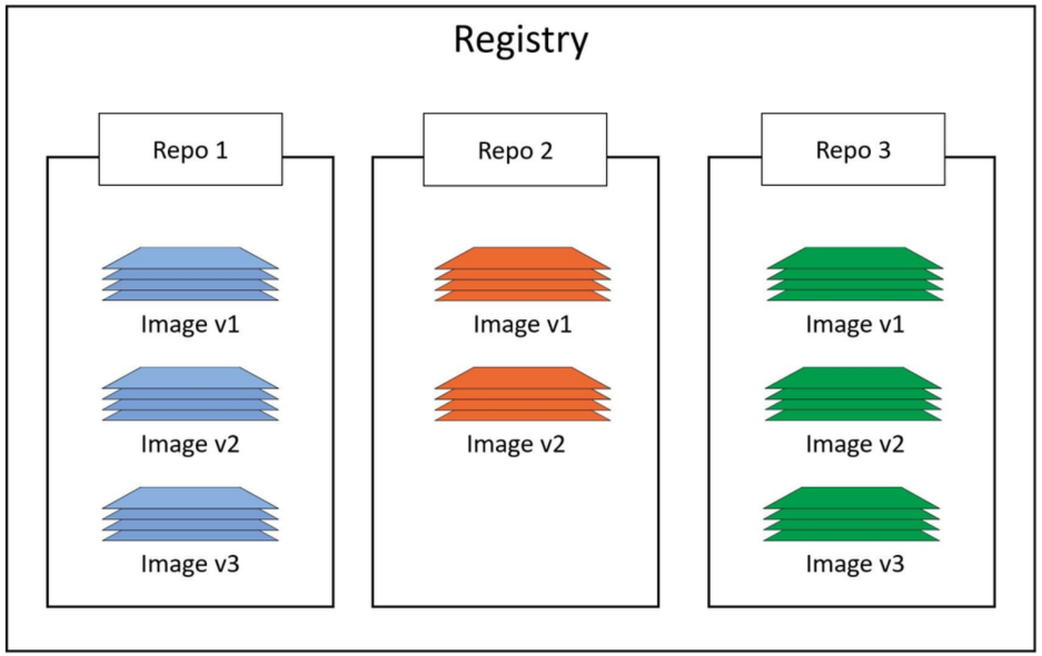
For example, Docker images do not ship with 6 different shells for you to choose from - they usually ship with a single minimalist shell, or no shell at all. They also don't contain a kernel - all containers running on a Docker host share access to the host's kernel.

For these reasons, we sometimes say images contain just enough operating system

Image registries

Docker images are stored in image registries. The most common registry is Docker Hub (https://hub.docker.com).

Other registries exist, including 3rd party registries and secure on-premises registries. However, the Docker client is opinionated and defaults to using Docker Hub.



Official & Unofficial registries

Official repositories contain images that have been vetted by Docker, Inc. This means they should contain up-to-date, high-quality code, that is secure, well-documented, and in-line with best practices.

Unofficial repositories can be like viking journey - you should not expect them to be safe, well-documented or built according to best practices. That’s not saying everything in unofficial repositories is bad! There’s some brilliant stuff in unofficial repositories. You just need to be very careful before trusting code from them.

Image naming & tagging

Addressing images from official repositories is as simple as giving the repository name and tag separated by a colon (:). The format for docker image pull , when working with an image from an official repository is: **docker image pull <repository>:<tag>**

```
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker image pull redis
Using default tag: latest
latest: Pulling from library/redis
afb6ec6fdc1c: Pull complete
608641ee4c3f: Pull complete
668ab9e1f4bc: Pull complete
78a12698914e: Pull complete
d056855f4300: Pull complete
618fdf7d0dec: Pull complete
Digest: sha256:ec277acf143340fa338f0b1a9b2f23632335d2096940d8e754474e21476eae32
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
deepone             latest              0d34f8b9483d        14 hours ago        279MB
alpine              latest              a24bb4013296        5 days ago          5.57MB
redis               latest              36304d3b4540        6 days ago          104MB
centos              latest              470671670cac        4 months ago        237MB
[root@ip-172-31-25-230 ~]#
```

Couple of things to remember:

if you do not specify an image tag after the repository name, Docker will assume you are referring to the image tagged as latest .

The latest tag doesn’t have any powers. It does not guarantee it is the most recent image in arepository.

Pulling images from an unofficial repository is essentially the same - you just need to prepend the repository name with a Docker Hub username or organization name.

Here we are pulling webserver v1 image from lovelearnlinux repository owned by Network Nuts.

```
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker pull lovelearnlinux/webserver:v1
v1: Pulling from lovelearnlinux/webserver
8a29a15cefae: Already exists
eabe6c48556e: Pull complete  <--
dd4435497983: Pull complete
Digest: sha256:ac9ab88604c58e12ada773e284f2f19d7e7a1564f741b2a0cc7f120ad8740b88
Status: Downloaded newer image for lovelearnlinux/webserver:v1
docker.io/lovelearnlinux/webserver:v1
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker images
REPOSITORY                TAG                 IMAGE ID            CREATED             SIZE
deepone                   latest              0d34f8b9483d        14 hours ago        279MB
alpine                    latest              a24bb4013296        5 days ago          5.57MB
redis                     latest              36304d3b4540        6 days ago          104MB
lovelearnlinux/webserver  v1                  083524bb2f84        5 weeks ago         277MB
centos                    latest              470671670cac        4 months ago        237MB
[root@ip-172-31-25-230 ~]#
```

I want you to remember the layer numbers , marked with arrow.

Here we are pulling webserver v1 image from lovelearnlinux repository owned by Network Nuts. Pull all of the images in a repository by adding the -a flag to them docker image pull command.

```
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker pull lovelearnlinux/webserver:v1
v1: Pulling from lovelearnlinux/webserver
8a29a15cefae: Already exists
eabe6c48556e: Pull complete    <--
dd4435497983: Pull complete
Digest: sha256:ac9ab88604c58e12ada773e284f2f19d7e7a1564f741b2a0cc7f120ad8740b88
Status: Downloaded newer image for lovelearnlinux/webserver:v1
docker.io/lovelearnlinux/webserver:v1
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker images
REPOSITORY                  TAG                 IMAGE ID            CREATED             SIZE
deepone                     latest              0d34f8b9483d        14 hours ago        279MB
alpine                      latest              a24bb4013296        5 days ago          5.57MB
redis                       latest              36304d3b4540        6 days ago          104MB
lovelearnlinux/webserver    v1                  083524bb2f84        5 weeks ago         277MB
centos                      latest              470671670cac        4 months ago        237MB
[root@ip-172-31-25-230 ~]#
```

I want you to remember the layer numbers , marked with arrow.

Image filtering on docker host

Docker provides the --filter flag to filter the list of images returned by docker image ls .
Docker currently supports the following filters:

- dangling: Accepts true or false , and returns only dangling images (true), or non-dangling images (false).
- before: Requires an image name or ID as argument, & returns all images created before it.
- since: Same as above, but returns images created after the specified image.
- label: Filters images based on the presence of a label or label and value. The docker image ls command does not display labels in its output

Example of using reference to display only images tagged as “latest”.

```
[root@ip-172-31-25-230 ~]# docker images ls --filter=reference="*:latest"
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
deepone             latest              0d34f8b9483d        15 hours ago        279MB
alpine              latest              a24bb4013296        5 days ago          5.57MB
redis               latest              36304d3b4540        6 days ago          104MB
centos              latest              470671670cac        4 months ago        237MB
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]#
```

Use the --format flag to format output using Go templates. For example, the following command will only return the size property of images on a Docker host.

```
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker image ls --format "{{.Size}}"
279MB
5.57MB
104MB
277MB
237MB
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]#
```

Show all images, but only display repo, tag and size.

```
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker image ls --format "{{.Repository}}: {{.Tag}}: {{.Size}}"
deepone: latest: 279MB
alpine: latest: 5.57MB
redis: latest: 104MB
lovelearnlinux/webserver: v1: 277MB
centos: latest: 237MB
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]#
```

Image searching on docker hub

The docker search command lets you search Docker Hub from the CLI. You can pattern match against strings in the “NAME” field, and filter output based on any of the returned columns.

In its simplest form, it searches for all repos containing a certain string in the “NAME” field. For example, the following command searches for all repos with “lovelearnlinux” in the “NAME” field.

```
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker search lovelearnlinux
NAME                          DESCRIPTION         STARS               OFFICIAL
lovelearnlinux/webserver                          0
lovelearnlinux/stress                             0
lovelearnlinux/busybox                            0
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]#
```

Use --filter "is-official=true" so that only official repos are displayed

```
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker search centos --filter "is-official=true"
NAME                DESCRIPTION                     STARS               OFFICIAL            AUTOMATED
centos              The official build of CentOS.   6032                [OK]
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker search python --filter "is-official=true"
NAME                DESCRIPTION                                     STARS               OFFICIAL
python              Python is an interpreted, interactive, objec…   5235                [OK]
django              Django is a free web application framework, …   966                 [OK]
pypy                PyPy is a fast, compliant alternative implem…   240                 [OK]
hylang              Hy is a Lisp dialect that translates express…   28                  [OK]
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]#
```

Show repos with automated builds

```
[root@ip-172-31-25-230 ~]# docker search python --filter "is-automated=true"
NAME                              DESCRIPTION                                     STARS
joyzoursky/python-chromedriver    Python with Chromedriver, for running automa…   49
nikolaik/python-nodejs            Python with Node.js                             46
bitnami/python                    Bitnami Python Docker Image                     6
publicisworldwide/python-conda    Basic Python environments with Conda.           6
dockershelf/python                Repository for docker images of Python. Test…   5
d3fk/python_in_bottle             Simple python:alpine completed by Bottle+Req…   2
komand/python-plugin              DEPRECATED: Komand Python SDK                   2
ccitest/python                    CircleCI test images for Python                 0
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]#
```

Use --filter "is-official=true" so that only official repos are displayed

```
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker search centos --filter "is-official=true"
NAME                DESCRIPTION                     STARS               OFFICIAL            AUTOMATED
centos              The official build of CentOS.   6032                [OK]
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker search python --filter "is-official=true"
NAME                DESCRIPTION                                     STARS               OFFICIAL
python              Python is an interpreted, interactive, objec…   5235                [OK]
django              Django is a free web application framework, …   966                 [OK]
pypy                PyPy is a fast, compliant alternative implem…   240                 [OK]
hylang              Hy is a Lisp dialect that translates express…   28                  [OK]
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]#
```

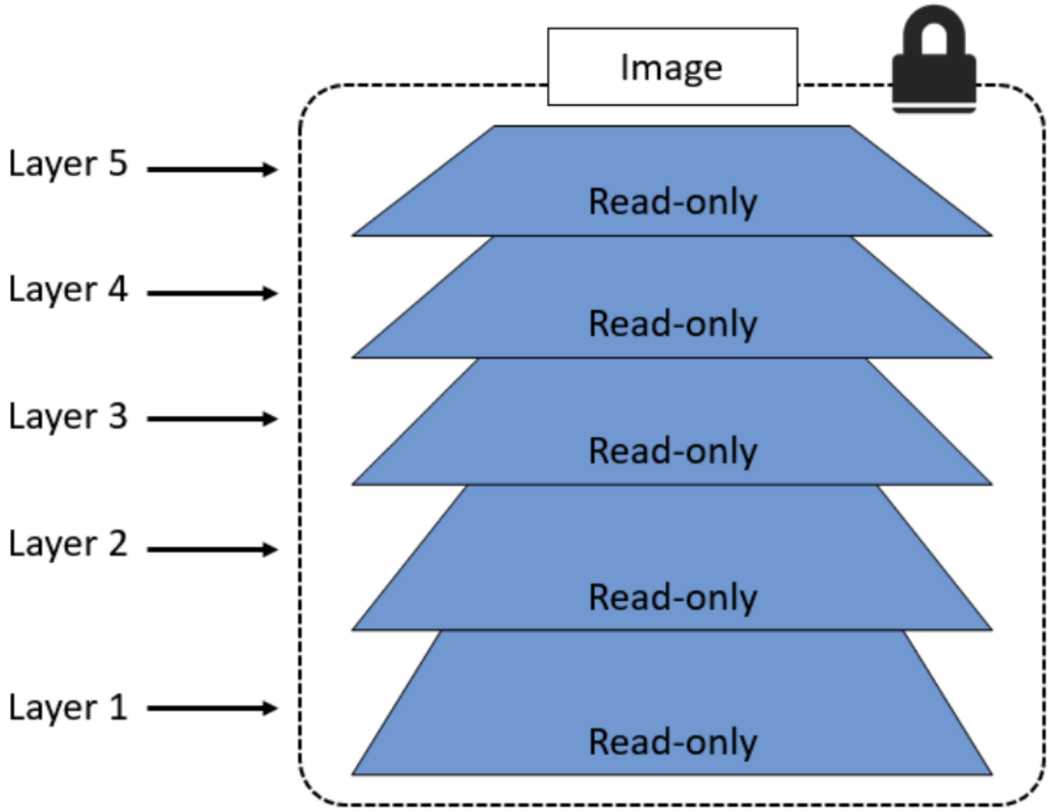
Show repos with automated builds

```
[root@ip-172-31-25-230 ~]# docker search python --filter "is-automated=true"
NAME                              DESCRIPTION                                     STARS
joyzoursky/python-chromedriver    Python with Chromedriver, for running automa…   49
nikolaik/python-nodejs            Python with Node.js                             46
bitnami/python                    Bitnami Python Docker Image                     6
publicisworldwide/python-conda    Basic Python environments with Conda.           6
dockershelf/python                Repository for docker images of Python. Test…   5
d3fk/python_in_bottle             Simple python:alpine completed by Bottle+Req…   2
komand/python-plugin              DEPRECATED: Komand Python SDK                   2
ccitest/python                    CircleCI test images for Python                 0
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]#
```

By default, Docker will only display 25 lines of results. However, you can use the --limit flag to increase that to a maximum of 100.

Images & layers

A Docker image is just a bunch of loosely-connected read-only layers. Docker takes care of stacking these layers and representing them as a single unified object.



There are a few ways to see and inspect the layers that make up an image, and we've already seen one of them.

```
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker pull lovelearnlinux/webserver:v1
v1: Pulling from lovelearnlinux/webserver
8a29a15cefae: Already exists
eabe6c48556e: Pull complete
dd4435497983: Pull complete
Digest: sha256:ac9ab88604c58e12ada773e284f2f19d7e7a1564f741b2a0cc7f120ad8740b88
Status: Downloaded newer image for lovelearnlinux/webserver:v1
docker.io/lovelearnlinux/webserver:v1
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker images
REPOSITORY                TAG                 IMAGE ID            CREATED             SIZE
deepone                   latest              0d34f8b9483d        14 hours ago        279MB
alpine                    latest              a24bb4013296        5 days ago          5.57MB
redis                     latest              36304d3b4540        6 days ago          104MB
lovelearnlinux/webserver  v1                  083524bb2f84        5 weeks ago         277MB
centos                    latest              470671670cac        4 months ago        237MB
[root@ip-172-31-25-230 ~]#
```

Each line in the output above that ends with "Pull complete" represents a layer in the image that was pulled. This image has 3 layers.

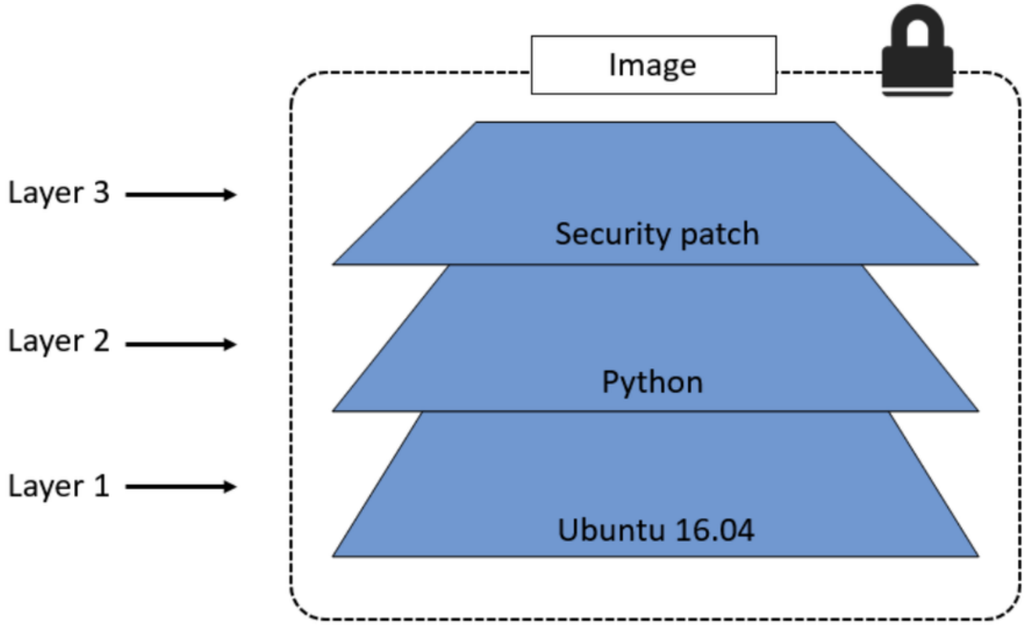
Another way to see the layers of an image is to inspect the image with the docker image inspect command.

 # docker image inspect lovelearnlinux/webserver:v1

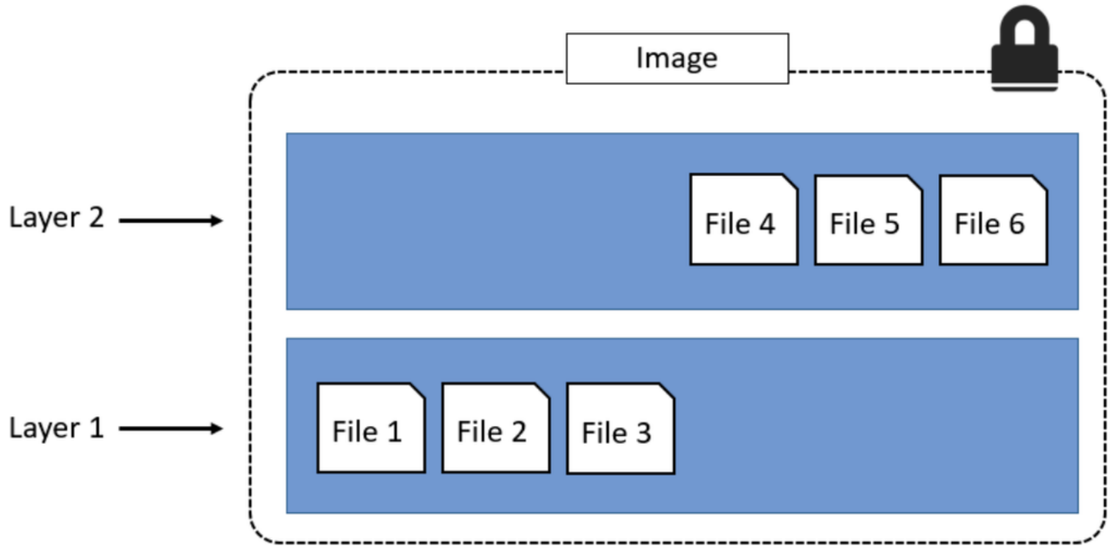
```
"RootFS": {
    "Type": "layers",
    "Layers": [
        "sha256:0683de2821778aa9546bf3d3e6944df779daba1582631b7ea3517bb36f9e4007",
        "sha256:f118e4e5bbe5c09db2b6ed603236b83e5de2a2e8c664c31cc4ce58e41f2b0727",
        "sha256:2a1e69ed439c6f0cb825a1cc8b52982b98cdcdbcf4f4f85e69f0eb3892b19607"
    ]
},
```

All Docker images start with a base layer, and as changes are made and new content is added, new layers are added on top.

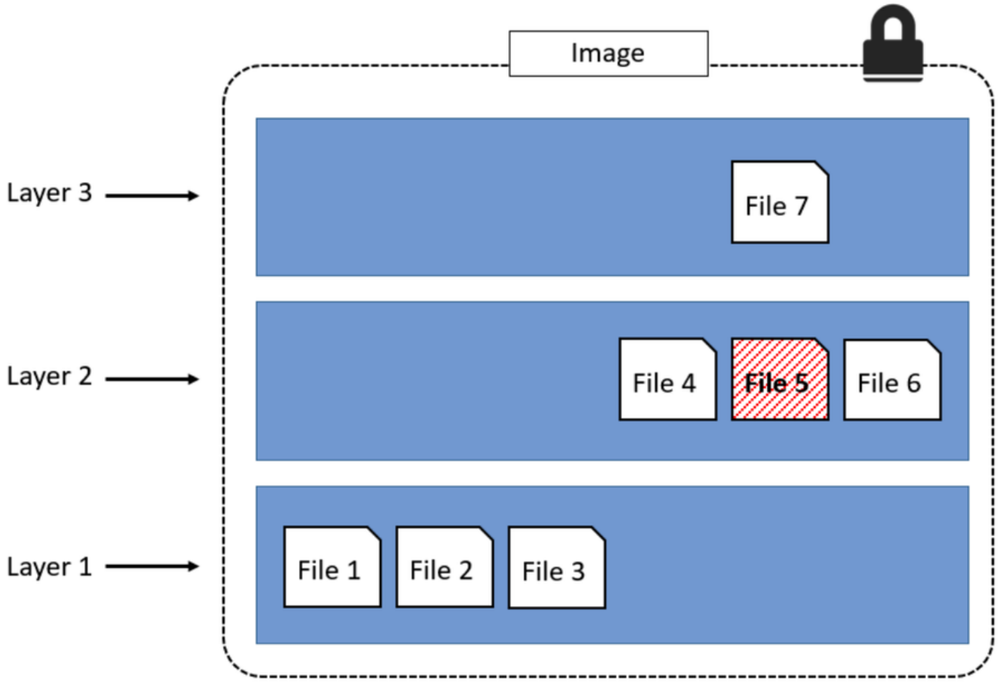
You might create a new image based off Ubuntu Linux 16.04. This would be your image's first layer. Then later if you add python, this would be added as a second layer on top of the base layer. If you then added a security patch, this would be added as a third layer at the top. Your image would now have three layers.



Remember, that as additional layers are added, the image is always the combination of all layers. Take a simple example of two layers. Each layer has 3 files, but the overall image has 6 files as it is the combination of both layers.



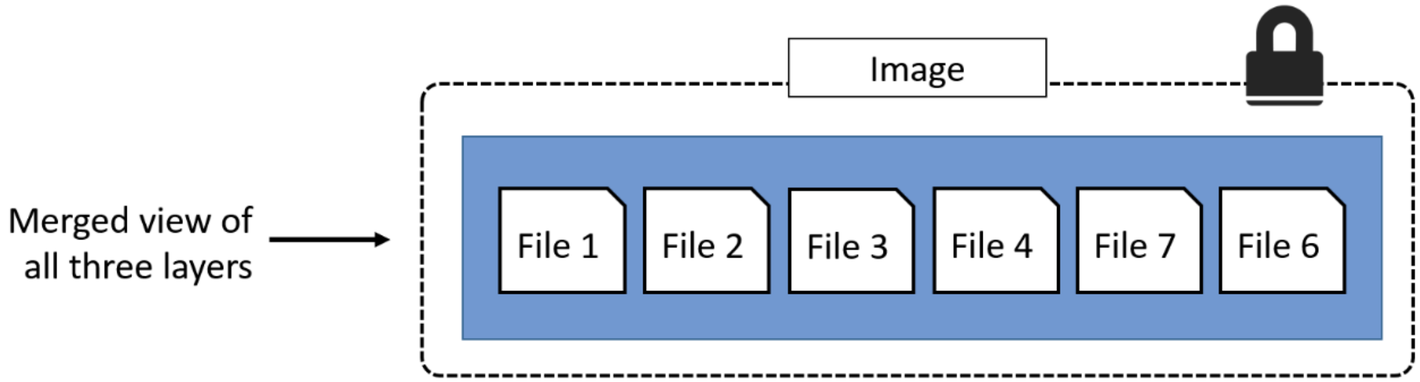
In this case of three-layered image. The overall image only presents 6 files in the unified view. This is because file 7 in the top layer is an updated version of file 5 directly below (inline). In this situation, the file in the higher layer obscures the file directly below it. This allows updated versions of files to be added as new layers to the image.



Docker employs a storage driver (snapshotter in newer versions) that is responsible for stacking layers and presenting them as a single unified filesystem. Examples of storage drivers on Linux include AUFS , overlay2 , devicemapper , btrfs and zfs .

The only driver supported by Docker on Windows is windowsfilter , which implements layering and CoW on top of NTFS.

The same 3-layer image as it will appear to the system. I.e. all three layers stacked and merged, giving a single unified view.



Sharing image layers

Multiple images can, and do, share layers. This leads to efficiencies in space & performance. Let’s do docker image pull command with the -a flag that we ran previously to pull all tagged images in the lovelearnlinux/webserver repository.

```
[root@ip-172-31-25-230 ~]# docker images
REPOSITORY                  TAG                 IMAGE ID            CREATED             SIZE
deepone                     latest              0d34f8b9483d        16 hours ago        279MB
alpine                      latest              a24bb4013296        5 days ago          5.57MB
redis                       latest              36304d3b4540        6 days ago          104MB
lovelearnlinux/webserver    v1                  083524bb2f84        5 weeks ago         277MB
centos                      latest              470671670cac        4 months ago        237MB
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker image pull -a lovelearnlinux/webserver
v1: Pulling from lovelearnlinux/webserver
Digest: sha256:ac9ab88604c58e12ada773e284f2f19d7e7a1564f741b2a0cc7f120ad8740b88
v2: Pulling from lovelearnlinux/webserver
8a29a15cefae: Already exists
eabe6c48556e: Already exists
b543811563e2: Pull complete
Digest: sha256:eea05b35b5159ed4e825935df2ed336317693eee40d568d2b4bee9d3ba4b7005
Status: Downloaded newer image for lovelearnlinux/webserver
docker.io/lovelearnlinux/webserver
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker images
REPOSITORY                  TAG                 IMAGE ID            CREATED             SIZE
deepone                     latest              0d34f8b9483d        16 hours ago        279MB
alpine                      latest              a24bb4013296        5 days ago          5.57MB
redis                       latest              36304d3b4540        6 days ago          104MB
lovelearnlinux/webserver    v2                  f9386045a7d3        5 weeks ago         277MB
lovelearnlinux/webserver    v1                  083524bb2f84        5 weeks ago         277MB
centos                      latest              470671670cac        4 months ago        237MB
[root@ip-172-31-25-230 ~]#
```

Notice the lines ending in Already exists .

These lines tell us that Docker is smart enough recognize when it’s being asked to pull an image layer that it already has a copy of. In this example, Docker pulled the image tagged as latest first. Then, when it pulled the v1 and v2 images, it noticed that it already had some of the layers that make up those images. This happens because both images in this repository are almost identical, and therefore share many layers.

Pulling image using digest

Normally we pull images by tag. But it has a problem — tags are mutable! This means it's possible to accidentally tag an image with the wrong tag. Sometimes it's even possible to tag an image with the same tag as an existing, but different, image. This can cause problems!

Docker 1.10 introduced a new content addressable storage model. As part of this new model, all images now get a cryptographic content hash. We will refer this hash as the digest.

Because the digest is a hash of the contents of the image, it is not possible to change the contents of the image without the digest also changing. This means digests are immutable.

This helps avoid the problem.

```
[root@ip-172-31-25-230 ~]# docker images
REPOSITORY                  TAG                 IMAGE ID            CREATED             SIZE
deepone                     latest              0d34f8b9483d        17 hours ago        279MB
alpine                      latest              a24bb4013296        5 days ago          5.57MB
redis                       latest              36304d3b4540        6 days ago          104MB
lovelearnlinux/webserver    v2                  f9386045a7d3        5 weeks ago         277MB
lovelearnlinux/webserver    v1                  083524bb2f84        5 weeks ago         277MB
centos                      latest              470671670cac        4 months ago        237MB
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker image ls --digests alpine
REPOSITORY          TAG                 DIGEST
 CREATED             SIZE
alpine              latest              sha256:185518070891758909c9f839cf4ca393ee977ac378609f700f60a771a2dfe321
 5 days ago          5.57MB
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]#
```

Now remote the image and try pulling it using digest / hash.

```
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker image ls --digests alpine
REPOSITORY          TAG                 DIGEST                                                                    IMAGE ID
 CREATED             SIZE
alpine              latest              sha256:18551807089175890 9c9f839cf4ca393ee977ac378609f700f60a771a2dfe321   a24bb4013296
 5 days ago          5.57MB
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker image rm alpine
Untagged: alpine:latest
Untagged: alpine@sha256:18551807089175890 9c9f839cf4ca393ee977ac378609f700f60a771a2dfe321
Deleted: sha256:a24bb4013296f61e89ba57005a7b3e52274d8edd3ae2077d04395f806b63d83e
Deleted: sha256:50644c29ef5a27c9a40c393a73ece2479de78325cae7d762ef3cdc19bf42dd0a
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker image pull alpine@sha256:18551807089175890 9c9f839cf4ca393ee977ac378609f700f60a771a2dfe321
sha256:18551807089175890 9c9f839cf4ca393ee977ac378609f700f60a771a2dfe321: Pulling from library/alpine
df20fa9351a1: Pull complete
Digest: sha256:18551807089175890 9c9f839cf4ca393ee977ac378609f700f60a771a2dfe321
Status: Downloaded newer image for alpine@sha256:18551807089175890 9c9f839cf4ca393ee977ac378609f700f60a771a2dfe321
docker.io/library/alpine@sha256:18551807089175890 9c9f839cf4ca393ee977ac378609f700f60a771a2dfe321
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]#
```

Deleting Images

Deleting an image will remove the image and all of its layers from your Docker host. This means it will no longer show up in docker image ls commands, and all directories on the Docker host containing the layer data will be deleted.

However, if an image layer is shared by more than one image, that layer will not be deleted until all images that reference it have been deleted.

```
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker image rm redis:latest
Untagged: redis:latest
Untagged: redis@sha256:ec277acf143340fa338f0b1a9b2f23632335d2096940d8e754474e21476eae32
Deleted: sha256:36304d3b4540c5143673b2cefaba583a0426b57c709b5a35363f96a3510058cd
Deleted: sha256:0a0f29e43c3a5555d675d253ff51d73e4d238bd558f11ae9b63d2a2a14251b36
Deleted: sha256:529a5b3d7258f27ee4d9aec07767e5bc40219f15406224945406b185a1225fb2
Deleted: sha256:0dafda0fc2be201f1695d0bd0c7ac81c5faf8974538f6e3c0d28021f1a258bda
Deleted: sha256:f65b9f3331a36179d24fde0f222c292349b30a7c0455e832491833b06f20a4eb
Deleted: sha256:ecbe43fbdb4faf2f576ef25feaed19e949912e4ffc3b2a1f5dee9a3fd52128c4
Deleted: sha256:ffc9b21953f4cd7956cdf532a5db04ff0a2daa7475ad796f1bad58cfbaf77a07
[root@ip-172-31-25-230 ~]#
```

If the image you are trying to delete is in use by a running container you will not be able to delete it. Stop and delete any containers before trying the delete operation again.

A handy shortcut for deleting all images on a Docker host is to run the docker image rm command and pass it a list of all image IDs on the system by calling docker image ls with the -q flag.

```
[root@ip-172-31-25-230 ~]# docker image ls -q
0d34f8b9483d
a24bb4013296
f9386045a7d3
083524bb2f84
470671670cac
[root@ip-172-31-25-230 ~]#
[root@ip-172-31-25-230 ~]# docker image rm $(docker image ls -q) -f
Untagged: deepone:latest
Deleted: sha256:0d34f8b9483d536e87009e833af3434dfe501b8bc74c3ea607f69f05661d29d2
Deleted: sha256:035f2c66fa85d4ba5f9b37f5d72a3fd5f739979bc75e69c40683b220b8f6e690
Deleted: sha256:9fee0d7bd186d4f558780ba29f62020eb302cf15ca5a0b4914358a41f6d3e276
Deleted: sha256:62777e266162323422940b6042fd64fd42f851a34b7de91448eea0b5151db992
Deleted: sha256:693789c9030e0ce691a38a0ee312c9782b5026a82f001d98cf0e817c6dfa3b22
Deleted: sha256:576b5809a5940ab1feb847c9f8096960d8b1d9bea062b8c7bab3e99118f08afb
Deleted: sha256:baa0c1672c77d4bc377c9a147e0239e5c666b70e472621a3b2ccb8bd33606d5d
Untagged: alpine@sha256:185518070891758909c9f839cf4ca393ee977ac378609f700f60a771a2dfe321
Deleted: sha256:a24bb4013296f61e89ba57005a7b3e52274d8edd3ae2077d04395f806b63d83e
```

WHAT WE LEARNED

Like virtual machine templates and are used to start containers. Under the hood they are made up one or more read-only layers, that when stacked together, make up the overall image.

The docker image pull command to pull some images into our Docker host’s local registry.

Image naming, official and unofficial repos, layering, sharing, and digest