docker images

Estimated reading time: 9 minutes

Description

List images

Usage

```
docker images [OPTIONS] [REPOSITORY[:TAG]]
```

Options

Name, shorthand	Default	Description
`--all , -a`		Show all images (default hides intermediate images)
`--digests`		Show digests
`--filter , -f`		Filter output based on conditions provided
`--format`		Pretty-print images using a Go template
`--no-trunc`		Don’t truncate output
`--quiet , -q`		Only show numeric IDs

Parent command

Command	Description

Command	Description
docker (https://docs.docker.com/engine/reference/commandline/docker)	The base command for the Docker CLI.

Extended description

The default `docker images` will show all top level images, their repository and tags, and their size.

Docker images have intermediate layers that increase reusability, decrease disk usage, and speed up `docker build` by allowing each step to be cached. These intermediate layers are not shown by default.

The `SIZE` is the cumulative space taken up by the image and all its parent images. This is also the disk space used by the contents of the Tar file created when you `docker save` an image.

An image will be listed more than once if it has multiple repository names or tags. This single image (identifiable by its matching `IMAGE ID`) uses up the `SIZE` listed only once.

Examples

List the most recently created images

```
$ docker images

REPOSITORY                TAG                 IMAGE ID
CREATED             SIZE
<none>                    <none>              77af4d6b9913
19 hours ago        1.089 GB
committ                   latest              b6fa739cedf5
19 hours ago        1.089 GB
<none>                    <none>              78a85c484f71
19 hours ago        1.089 GB
docker                    latest              30557a29d5ab
20 hours ago        1.089 GB
<none>                    <none>              5ed6274db6ce
24 hours ago        1.089 GB
postgres                  9                   746b819f315e
4 days ago          213.4 MB
postgres                  9.3                 746b819f315e
4 days ago          213.4 MB
postgres                  9.3.5               746b819f315e
4 days ago          213.4 MB
postgres                  latest              746b819f315e
4 days ago          213.4 MB
```

List images by name and tag

The `docker images` command takes an optional `[REPOSITORY[:TAG]]` argument that restricts the list to images that match the argument. If you specify `REPOSITORY` but no `TAG`, the `docker images` command lists all images in the given repository.

For example, to list all images in the "java" repository, run this command :

```
$ docker images java

REPOSITORY          TAG                 IMAGE ID            CREATE
D             SIZE
java                8                   308e519aac60        6 days
 ago          824.5 MB
java                7                   493d82594c15        3 mont
hs ago        656.3 MB
java                latest              2711b1d6f3aa        5 mont
hs ago        603.9 MB
```

The `[REPOSITORY[:TAG]]` value must be an “exact match”. This means that, for example, `docker images jav` does not match the image `java` .

If both `REPOSITORY` and `TAG` are provided, only images matching that repository and tag are listed. To find all local images in the “java” repository with tag “8” you can use:

```
$ docker images java:8

REPOSITORY          TAG                 IMAGE ID            CREATE
D             SIZE
java                8                   308e519aac60        6 days
 ago          824.5 MB
```

If nothing matches `REPOSITORY[:TAG]` , the list is empty.

```
$ docker images java:0

REPOSITORY          TAG                 IMAGE ID            CREATE
D             SIZE
```

List the full length image IDs

```
$ docker images --no-trunc

REPOSITORY                    TAG                 IMAGE ID
                                                         CREATED
            SIZE
<none>                        <none>              sha256:77af4d6b9
913e693e8d0b4b294fa62ade6054e6b2f1ffb617ac955dd63fb0182   19 hours
 ago        1.089 GB
committest                    latest              sha256:b6fa739ce
df5ea12a620a439402b6004d057da800f91c7524b5086a5e4749c9f   19 hours
 ago        1.089 GB
<none>                        <none>              sha256:78a85c484
f71509adeaace20e72e941f6bdd2b25b4c75da8693efd9f61a37921   19 hours
 ago        1.089 GB
docker                        latest              sha256:30557a29d
5abc51e5f1d5b472e79b7e296f595abcf19fe6b9199dbbc809c6ff4   20 hours
 ago        1.089 GB
<none>                        <none>              sha256:0124422dd
9f9cf7ef15c0617cda3931ee68346455441d66ab8bdc5b05e9fdce5   20 hours
 ago        1.089 GB
<none>                        <none>              sha256:18ad6fad3
40262ac2a636efd98a6d1f0ea775ae3d45240d3418466495a19a81b   22 hours
 ago        1.082 GB
<none>                        <none>              sha256:f9f1e2635
2f0a3ba6a0ff68167559f64f3e21ff7ada60366e2d44a04befd1d3a   23 hours
 ago        1.089 GB
tryout                        latest              sha256:2629d1fa0
b81b222fca63371ca16cbf6a0772d07759ff80e8d1369b926940074   23 hours
 ago        131.5 MB
<none>                        <none>              sha256:5ed6274db
6ceb2397844896966ea239290555e74ef307030ebb01ff91b1914df   24 hours
 ago        1.089 GB
```

List image digests

Images that use the v2 or later format have a content-addressable identifier called a `digest`. As long as the input used to generate the image is unchanged, the digest value is predictable. To list image digest values, use the `--digests` flag:

```
$ docker images --digests
REPOSITORY                          TAG                 DIGEST
                                                                IMA
GE ID            CREATED             SIZE
localhost:5000/test/busybox         <none>              sha256:cbbf
2f9a99b47fc460d422812b6a5adff7dfee951d8fa2e4a98caa0382cfbdbf   498
6bf8c1536        9 weeks ago         2.43 MB
```

When pushing or pulling to a 2.0 registry, the `push` or `pull` command output includes the image digest. You can `pull` using a digest value. You can also reference by digest in `create`, `run`, and `rmi` commands, as well as the `FROM` image reference in a Dockerfile.

Filtering

The filtering flag (`-f` or `--filter`) format is of "key=value". If there is more than one filter, then pass multiple flags (e.g., `--filter "foo=bar" --filter "bif=baz"`)

The currently supported filters are:

- dangling (boolean - true or false)
- label (`label=<key>` or `label=<key>=<value>`)
- before (`<image-name>[:<tag>]`, `<image id>` or `<image@digest>`) - filter images created before given id or references
- since (`<image-name>[:<tag>]`, `<image id>` or `<image@digest>`) - filter images created since given id or references
- reference (pattern of an image reference) - filter images whose reference matches the specified pattern

SHOW UNTAGGED IMAGES (DANGLING)

```
$ docker images --filter "dangling=true"

REPOSITORY          TAG                 IMAGE ID            CREATE
D             SIZE
<none>              <none>              8abc22fbb042        4 week
s ago         0 B
<none>              <none>              48e5f45168b9        4 week
s ago         2.489 MB
<none>              <none>              bf747efa0e2f        4 week
s ago         0 B
<none>              <none>              980fe10e5736        12 wee
ks ago        101.4 MB
<none>              <none>              dea752e4e117        12 wee
ks ago        101.4 MB
<none>              <none>              511136ea3c5a        8 mont
hs ago        0 B
```

This will display untagged images that are the leaves of the images tree (not intermediary layers). These images occur when a new build of an image takes the `repo:tag` away from the image ID, leaving it as `<none>:<none>` or untagged. A warning will be issued if trying to remove an image when a container is presently using it. By having this flag it allows for batch cleanup.

You can use this in conjunction with `docker rmi ...` :

```
$ docker rmi $(docker images -f "dangling=true" -q)

8abc22fbb042
48e5f45168b9
bf747efa0e2f
980fe10e5736
dea752e4e117
511136ea3c5a
```

> **Note**: Docker warns you if any containers exist that are using these untagged images.

SHOW IMAGES WITH A GIVEN LABEL

The `label` filter matches images based on the presence of a `label` alone or a `label` and a value.

The following filter matches images with the `com.example.version` label

regardless of its value.

```
$ docker images --filter "label=com.example.version"

REPOSITORY          TAG                 IMAGE ID            CREATE
D              SIZE
match-me-1          latest              eeae25ada2aa        About
a minute ago   188.3 MB
match-me-2          latest              dea752e4e117        About
a minute ago   188.3 MB
```

The following filter matches images with the `com.example.version` label with the `1.0` value.

```
$ docker images --filter "label=com.example.version=1.0"

REPOSITORY          TAG                 IMAGE ID            CREATE
D              SIZE
match-me            latest              511136ea3c5a        About
a minute ago   188.3 MB
```

In this example, with the `0.1` value, it returns an empty set because no matches were found.

```
$ docker images --filter "label=com.example.version=0.1"
REPOSITORY          TAG                 IMAGE ID            CREATE
D              SIZE
```

FILTER IMAGES BY TIME

The `before` filter shows only images created before the image with given id or reference. For example, having these images:

```
$ docker images

REPOSITORY          TAG                 IMAGE ID            CREATE
D              SIZE
image1              latest              eeae25ada2aa        4 minu
tes ago        188.3 MB
image2              latest              dea752e4e117        9 minu
tes ago        188.3 MB
image3              latest              511136ea3c5a        25 min
utes ago       188.3 MB
```

Filtering with `before` would give:

```
$ docker images --filter "before=image1"

REPOSITORY          TAG                 IMAGE ID            CREATE
D              SIZE
image2              latest              dea752e4e117        9 minu
tes ago        188.3 MB
image3              latest              511136ea3c5a        25 min
utes ago       188.3 MB
```

Filtering with `since` would give:

```
$ docker images --filter "since=image3"
REPOSITORY          TAG                 IMAGE ID            CREATE
D              SIZE
image1              latest              eeae25ada2aa        4 minu
tes ago        188.3 MB
image2              latest              dea752e4e117        9 minu
tes ago        188.3 MB
```

FILTER IMAGES BY REFERENCE

The `reference` filter shows only images whose reference matches the specified pattern.

```
$ docker images

REPOSITORY          TAG                 IMAGE ID            CREATE
D             SIZE
busybox             latest              e02e811dd08f        5 week
s ago         1.09 MB
busybox             uclibc              e02e811dd08f        5 week
s ago         1.09 MB
busybox             musl                733eb3059dce        5 week
s ago         1.21 MB
busybox             glibc               21c16b6787c6        5 week
s ago         4.19 MB
```

Filtering with `reference` would give:

```
$ docker images --filter=reference='busy*:*libc'

REPOSITORY          TAG                 IMAGE ID            CREATE
D             SIZE
busybox             uclibc              e02e811dd08f        5 week
s ago         1.09 MB
busybox             glibc               21c16b6787c6        5 week
s ago         4.19 MB
```

Format the output

The formatting option (`--format`) will pretty print container output using a Go template.

Valid placeholders for the Go template are listed below:

Placeholder	Description
`.ID`	Image ID
`.Repository`	Image repository
`.Tag`	Image tag
`.Digest`	Image digest
`.CreatedSince`	Elapsed time since the image was created
`.CreatedAt`	Time when the image was created
`.Size`	Image disk size

When using the `--format` option, the `image` command will either output the data exactly as the template declares or, when using the `table` directive, will include column headers as well.

The following example uses a template without headers and outputs the `ID` and `Repository` entries separated by a colon for all images:

```
$ docker images --format "{{.ID}}: {{.Repository}}"

77af4d6b9913: <none>
b6fa739cedf5: committ
78a85c484f71: <none>
30557a29d5ab: docker
5ed6274db6ce: <none>
746b819f315e: postgres
746b819f315e: postgres
746b819f315e: postgres
746b819f315e: postgres
```

To list all images with their repository and tag in a table format you can use:

```
$ docker images --format "table {{.ID}}\t{{.Repository}}\t{{.Tag}}
"

IMAGE ID            REPOSITORY                TAG
77af4d6b9913        <none>                    <none>
b6fa739cedf5        committ                   latest
78a85c484f71        <none>                    <none>
30557a29d5ab        docker                    latest
5ed6274db6ce        <none>                    <none>
746b819f315e        postgres                  9
746b819f315e        postgres                  9.3
746b819f315e        postgres                  9.3.5
746b819f315e        postgres                  latest
```