

NAME

docker-image-ls - List images

SYNOPSIS

docker image ls [OPTIONS] [REPOSITORY[:TAG]]

DESCRIPTION

This command lists the images stored in the local Docker repository.

By default, intermediate images, used during builds, are not listed. Some of the output, e.g., image ID, is truncated, for space reasons. However the truncated image ID, and often the first few characters, are enough to be used in other Docker commands that use the image ID. The output includes repository, tag, image ID, date created and the virtual size.

The title REPOSITORY for the first title may seem confusing. It is essentially the image name. However, because you can tag a specific image, and multiple tags (image instances) can be associated with a single name, the name is really a repository for all tagged images of the same name. For example consider an image called ubuntu. It may be tagged with 24.04 or 22.04, etc. to manage different versions.

Filters

Filters the output based on these conditions:

- `dangling=(true|false)` - find unused images
- `label=` or `label==`
- `before=([:tag]||image@digest <mailto:image@digest>)`
- `since=([:tag]||image@digest <mailto:image@digest>)`
- `reference=(pattern of an image reference)`

Format

Format the output using the given Go template.

Valid placeholders:

.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

EXAMPLES**Listing the images**

To list the images in a local repository (not the registry) run:

```
docker image ls
```

The list will contain the image repository name, a tag for the image, and an image ID, when it was created and its virtual size. Columns: REPOSITORY, TAG, IMAGE ID, CREATED, and SIZE.

The **docker image ls** 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 image ls** command lists all images in the given repository.

```
docker image ls java
```

The **[REPOSITORY[:TAG]]** value must be an "exact match". This means that, for example, **docker image ls 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 image ls java:8
```

To get a verbose list of images which contains all the intermediate images used in builds use **-a**:

```
docker image ls -a
```

Previously, the **docker image ls** command supported the **--tree** and **--dot** arguments, which displayed different visualizations of the image data. Docker core removed this functionality in the 1.7 version. If you liked this functionality, you can still find it in the third-party **dockviz** tool: <https://github.com/justone/dockviz>.

Listing images in a desired format

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. You can use special characters like **\t** for inserting tab spacing between columns. The **json** directive outputs objects in JSON format.

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

```
docker image ls --format "{{.ID}}: {{.Repository}}"
77af4d6b9913: <none>
b6fa739cedf5: committ
78a85c484bad: ipbabbble
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 image ls --format "table {{.ID}}\t{{.Repository}}\t{{.Tag}}"
IMAGE ID          REPOSITORY          TAG
77af4d6b9913      <none>              <none>
b6fa739cedf5      committ             latest
78a85c484bad      ipbabbble           <none>
30557a29d5ab      docker              latest
5ed6274db6ce      <none>              <none>
746b819f315e      postgres            9
746b819f315e      postgres            9.3
```

746b819f315e	postgres	9.3.5
746b819f315e	postgres	latest

Valid template placeholders are listed above.

To list all images in JSON format you can use:

```
docker image ls --format json
{"Containers":"N/A","CreatedAt":"2021-01-18 11:29:06 +0100 CET","CreatedSince":"24 h
{"Containers":"N/A","CreatedAt":"2021-01-18 11:24:48 +0100 CET","CreatedSince":"24 h
{"Containers":"N/A","CreatedAt":"2021-01-18 10:43:44 +0100 CET","CreatedSince":"24 h
```

Listing only the shortened image IDs

Listing just the shortened image IDs. This can be useful for some automated tools.

```
docker image ls -q
```

OPTIONS

-a, --all[=false] Show all images (default hides intermediate images)

--digests[=false] Show digests

-f, --filter= Filter output based on conditions provided

--format="" Format output using a custom template: 'table': Print output in table format with column headers (default) 'table TEMPLATE': Print output in table format using the given Go template 'json': Print in JSON format 'TEMPLATE': Print output using the given Go template. Refer to <https://docs.docker.com/go/formatting/> for more information about formatting output with templates

--no-trunc[=false] Don't truncate output

-q, --quiet[=false] Only show image IDs

--tree[=false] List multi-platform images as a tree (EXPERIMENTAL)

SEE ALSO

docker-image(1)