Docker object labels

Estimated reading time: 3 minutes

Labels are a mechanism for applying metadata to Docker objects, including:

- Images
- Containers
- Local daemons
- Volumes
- Networks
- Swarm nodes
- Swarm services

You can use labels to organize your images, record licensing information, annotate relationships between containers, volumes, and networks, or in any way that makes sense for your business or application.

Label keys and values

A label is a key-value pair, stored as a string. You can specify multiple labels for an object, but each key-value pair must be unique within an object. If the same key is given multiple values, the most-recently-written value overwrites all previous values.

Key format recommendations

A label *key* is the left-hand side of the key-value pair. Keys are alphanumeric strings which may contain periods (`.`) and hyphens (`-`). Most Docker users use images created by other organizations, and the following guidelines help to prevent inadvertent duplication of labels across objects, especially if you plan to use labels as a mechanism for automation.

- Authors of third-party tools should prefix each label key with the reverse DNS notation of a domain they own, such as `com.example.some-label` .
- Do not use a domain in your label key without the domain owner’s permission.
- The `com.docker.*` , `io.docker.*` , and `org.dockerproject.*` namespaces are reserved by Docker for internal use.

- Label keys should begin and end with a lower-case letter and should only contain lower-case alphanumeric characters, the period character (`.`), and the hyphen character (`-`). Consecutive periods or hyphens are not allowed.
- The period character (`.`) separates namespace “fields”. Label keys without namespaces are reserved for CLI use, allowing users of the CLI to interactively label Docker objects using shorter typing-friendly strings.

These guidelines are not currently enforced and additional guidelines may apply to specific use cases.

Value guidelines

Label values can contain any data type that can be represented as a string, including (but not limited to) JSON, XML, CSV, or YAML. The only requirement is that the value be serialized to a string first, using a mechanism specific to the type of structure. For instance, to serialize JSON into a string, you might use the `JSON.stringify()` JavaScript method.

Since Docker does not deserialize the value, you cannot treat a JSON or XML document as a nested structure when querying or filtering by label value unless you build this functionality into third-party tooling.

Manage labels on objects

Each type of object with support for labels has mechanisms for adding and managing them and using them as they relate to that type of object. These links provide a good place to start learning about how you can use labels in your Docker deployments.

Labels on images, containers, local daemons, volumes, and networks are static for the lifetime of the object. To change these labels you must recreate the object. Labels on swarm nodes and services can be updated dynamically.

- Images and containers
    - Adding labels to images (https://docs.docker.com/engine/reference/builder/#label)
    - Overriding a container’s labels at runtime (https://docs.docker.com/engine/reference/commandline/run/#set-metadata-on-container--l---label---label-file)
    - Inspecting labels on images or containers (https://docs.docker.com/engine/reference/commandline/inspect/)
    - Filtering images by label (https://docs.docker.com/engine/reference/commandline/images/#filtering)
    - Filtering containers by label (https://docs.docker.com/engine/reference/commandline/ps/#filtering)

- Local Docker daemons
    - Adding labels to a Docker daemon at runtime (https://docs.docker.com/engine/reference/commandline/dockerd/)
    - Inspecting a Docker daemon's labels (https://docs.docker.com/engine/reference/commandline/info/)
- Volumes
    - Adding labels to volumes (https://docs.docker.com/engine/reference/commandline/volume_create/)
    - Inspecting a volume's labels (https://docs.docker.com/engine/reference/commandline/volume_inspect/)
    - Filtering volumes by label (https://docs.docker.com/engine/reference/commandline/volume_ls/#filtering)
- Networks
    - Adding labels to a network (https://docs.docker.com/engine/reference/commandline/network_create/)
    - Inspecting a network's labels (https://docs.docker.com/engine/reference/commandline/network_inspect/)
    - Filtering networks by label (https://docs.docker.com/engine/reference/commandline/network_ls/#filtering)
- Swarm nodes
    - Adding or updating a swarm node's labels (https://docs.docker.com/engine/reference/commandline/node_update/#add-label-metadata-to-a-node)
    - Inspecting a swarm node's labels (https://docs.docker.com/engine/reference/commandline/node_inspect/)
    - Filtering swarm nodes by label (https://docs.docker.com/engine/reference/commandline/node_ls/#filtering)
- Swarm services
    - Adding labels when creating a swarm service (https://docs.docker.com/engine/reference/commandline/service_create/#set-metadata-on-a-service-l-label)
    - Updating a swarm service's labels (https://docs.docker.com/engine/reference/commandline/service_update/)
    - Inspecting a swarm service's labels (https://docs.docker.com/engine/reference/commandline/service_inspect/)
    - Filtering swarm services by label (https://docs.docker.com/engine/reference/commandline/service_ls/#filtering)

Usage (https://docs.docker.com/glossary/?term=Usage), user guide (https://docs.docker.com/glossary/?term=user guide), labels (https://docs.docker.com/glossary/?term=labels), metadata (https://docs.docker.com/glossary/?term=metadata), docker (https://docs.docker.com/glossary/?term=docker), documentation (https://docs.docker.com/glossary/?term=documentation), examples (https://docs.docker.com/glossary/?term=examples), annotating (https://docs.docker.com/glossary/?term=annotating)