Mirror the Docker Hub library

- Alternatives
- Gotcha
- Solution
- Using Registry Access Management (RAM) with a registry mirror
- How does it work?
  - What if the content changes on the Hub?
  - What about my disk?
- Run a Registry as a pull-through cache
  - Configure the cache
  - Configure the Docker daemon

If you have multiple instances of Docker running in your environment, such as multiple physical or virtual machines all running Docker, each daemon goes out to the internet and fetches an image it doesn't have locally, from the Docker repository. You can run a local registry mirror and point all your daemons there, to avoid this extra internet traffic.

> Docker Official Images are an intellectual property of Docker.

Alternatives

Alternatively, if the set of images you are using is well delimited, you can simply pull them manually and push them to a simple, local, private registry.

Furthermore, if your images are all built in-house, not using the Hub at all and relying entirely on your local registry is the simplest scenario.

Gotcha

It's currently not possible to mirror another private registry. Only the central Hub can be mirrored.

> Mirrors of Docker Hub are still subject to Docker's fair use policy.

Solution

The Registry can be configured as a pull through cache. In this mode a Registry responds to all normal docker pull requests but stores all content locally.

Using Registry Access Management (RAM) with a registry mirror

If Docker Hub access is restricted via your Registry Access Management (RAM) configuration, you will not be able to pull images originating from Docker Hub even if the images are available in your registry mirror.

You will encounter the following error:

```
Error response from daemon: Access to docker.io has been restricted by your administrators.
```

If you are unable to allow access to Docker Hub, you can manually pull from your registry mirror and optionally, retag the image. For example:

```
docker pull <your-registry-mirror>[:<port>]/library/busybox
docker tag <your-registry-mirror>[:<port>]/library/busybox:latest busybox:latest
```

How does it work?

The first time you request an image from your local registry mirror, it pulls the image from the public Docker registry and stores it locally before handing it back to you. On subsequent requests, the local registry mirror is able to serve the image from its own storage.

What if the content changes on the Hub?

When a pull is attempted with a tag, the Registry checks the remote to ensure if it has the latest version of the requested content. Otherwise, it fetches and caches the latest content.

What about my disk?

In environments with high churn rates, stale data can build up in the cache. When running as a pull through cache the Registry periodically removes old content to save disk space. Subsequent requests for removed content causes a remote fetch and local re-caching.

To ensure best performance and guarantee correctness the Registry cache should be configured to use the `filesystem` driver for storage.

Run a Registry as a pull-through cache

The easiest way to run a registry as a pull through cache is to run the official Registry image. At least, you need to specify `proxy.remoteurl` within `/etc/docker/registry/config.yml` as described in the following subsection.

Multiple registry caches can be deployed over the same back-end. A single registry cache ensures that concurrent requests do not pull duplicate data, but this property does not hold true for a registry cache cluster.

Configure the cache

To configure a Registry to run as a pull through cache, the addition of a `proxy` section is required to the config file.

To access private images on the Docker Hub, a username and password can be supplied.

```
proxy:
  remoteurl: https://registry-1.docker.io
  username: [username]
  password: [password]
```

If you specify a username and password, it's very important to understand that private resources that this user has access to Docker Hub is made available on your mirror. You must secure your mirror by implementing authentication if you expect these resources to stay private!

For the scheduler to clean up old entries, `delete` must be enabled in the registry configuration.

Configure the Docker daemon

Either pass the `--registry-mirror` option when starting `dockerd` manually, or edit `/etc/docker/daemon.json` and add the `registry-mirrors` key and value, to make the change persistent.

```
{
  "registry-mirrors": ["https://<my-docker-mirror-host>"]
}
```

Save the file and reload Docker for the change to take effect.

Some log messages that appear to be errors are actually informational messages.

Check the `level` field to determine whether the message is warning you about an error or is giving you information. For example, this log message is informational:

```
time="2017-06-02T15:47:37Z" level=info msg="error statting local store, serving from upstream: unknown blob" go.version=go1.7.4
```

It's telling you that the file doesn't exist yet in the local cache and is being pulled from upstream.

□□□ #2
□ □□□ □□□ 2026-10-31 19:19:24 UTC
□ □□□ □□□ 2026-10-31 19:19:25 UTC