Lab: Managing Docker Images

Introduction:

Docker images are the **building blocks** of containers in the Docker ecosystem. They are read-only templates that include all the necessary instructions for creating a container. Each image consists of a set of layers, each of which represents a change made to the file system of the container. These layers are stacked on top of each other to create the final image.

Images can be created from scratch or built on top of existing images. They can also be pulled from a remote repository, such as Docker Hub, and used to create new containers.

Because they are read-only, they are considered to be lightweight and portable, making them well suited for distributed and microservices-based applications. With an image you can also create many containers, each with the same configuration and dependencies, making it easy to run the same application in different environments or to replicate the same service multiple times.

Objective:

- **Search Image**
- **Pull an Image**
- **Pulling an image from different registry**
- **Cleanup**

1 Ensure that you have logged-in as **root** user with password as **linux**.

1.1 Let's **list** the images by executing the below command.

```
# docker image ls
```

Output:

```
[root@docker-engine ~]#docker image ls
REPOSITORY   TAG        IMAGE ID   CREATED   SIZE
```

1.2 Let's **search** image by executing the below command.

```
# docker search alpine
```

Output:

```
[root@docker-engine ~]#docker search alpine
NAME                              DESCRIPTION                                     STARS     OFFICIAL   AUTOMAT
ED
alpine                            A minimal Docker image based on Alpine Linux…   9705      [OK]
alpinelinux/docker-cli            Simple and lightweight Alpine Linux image wi…   7
alpinelinux/gitlab-runner         Alpine Linux gitlab-runner (supports more ar…   4
alpinelinux/alpine-gitlab-ci      Build Alpine Linux packages with Gitlab CI      3
alpinelinux/gitlab                Alpine Linux based Gitlab image                 2
grafana/alpine                    Alpine Linux with ca-certificates package in…   2
alpinelinux/gitlab-runner-helper  Helper image container gitlab-runner-helper …   2
alpinelinux/unbound                                                               1
alpinelinux/package-builder       Container to build packages for a repository    1
rancher/alpine-git                                                                1
alpinelinux/golang                Build container for golang based on Alpine L…   1
alpinelinux/darkhttpd                                                             1
alpinelinux/rsyncd                                                                0
alpinelinux/docker-abuild         Dockerised abuild                               0
alpinelinux/build-base            Base image suitable for building packages wi…   0
alpinelinux/mqtt-exec                                                             0
alpinelinux/docker-compose        docker-compose image based on Alpine Linux      0
alpinelinux/docker-alpine                                                         0
alpinelinux/git-mirror-syncd                                                      0
alpinelinux/ansible               Ansible in docker                               0
alpinelinux/mirror-status                                                         0
alpinelinux/alpine-docker-gitlab  Gitlab running on Alpine Linux                  0
alpinelinux/apkbuild-lint-tools   Tools for linting APKBUILD files in a CI env…   0
alpinelinux/alpine-drone-ci       Build Alpine Linux packages with drone CI       0
alpinelinux/aports-qa-bot         A Gitlab bot that gives feedback on aports m…   0
```

1.3 Let's search image by applying the **filter** by executing the below command.

```
# docker search alpine --filter=stars=100
```

Output:

```
[root@docker-engine ~]#docker search alpine --filter=stars=100
NAME      DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
alpine    A minimal Docker image based on Alpine Linux…   9656      [OK]
```

2 Let's **pull** an image by executing the below command.

```
# docker image pull alpine
```

Output:

```
[root@docker-engine ~]#docker image pull alpine
Using default tag: latest
latest: Pulling from library/alpine
8921db27df28: Pull complete
Digest: sha256:f271e74b17ced29b915d351685fd4644785c6d1559dd1f2d4189a5e851ef753a
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest
```

Note: By default, pull will get the Official latest image from the Docker hub.

2.1 Let's **list** the images by executing the below command.

```
# docker image ls
```

Output:

```
[root@docker-engine ~]#docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
alpine        latest    042a816809aa   4 weeks ago    7.05MB
```

3 Let's **pull** an image from a **specific repository** by executing the below command.

```
# docker image pull docker.io/alpine/git
```

Output:

```
[root@docker-engine ~]#docker image pull docker.io/alpine/git
Using default tag: latest
latest: Pulling from alpine/git
ca7dd9ec2225: Pull complete
eb878e0a08e4: Pull complete
b4f093b99828: Pull complete
Digest: sha256:66b210a97bc07bfd4019826bcd13a488b371a6cbe2630a4b37d23275658bd3f2
Status: Downloaded newer image for alpine/git:latest
docker.io/alpine/git:latest
```

3.1 Let's **pull** an image from a **specific repository** by executing the below command.

```
# docker image pull
quay.io/fluentd_elasticsearch/fluentd:v2.5.2
```

Output:

```
[root@docker-engine ~]#docker image pull quay.io/fluentd_elasticsearch/fluentd:v2.5.2
v2.5.2: Pulling from fluentd_elasticsearch/fluentd
fc7181108d40: Pull complete
549fa7c1306d: Pull complete
57be1f33cffb: Pull complete
87b7758c3481: Pull complete
a975c5d5269b: Pull complete
e9e2325ff3ce: Pull complete
a68d19d3fb0e: Pull complete
Digest: sha256:aec118bb3d1c4af358c1d495b14c12781c4ab5e8cfb455edb1ebd2e92750e31d
Status: Downloaded newer image for quay.io/fluentd_elasticsearch/fluentd:v2.5.2
quay.io/fluentd_elasticsearch/fluentd:v2.5.2
```

3.2 Let's **pull** an image from a **specific repository** by executing the below command.

```
# docker image pull registry.k8s.io/etcd:3.5.6-0
```

Output:

```
[root@docker-engine ~]#docker image pull registry.k8s.io/etcd:3.5.6-0
3.5.6-0: Pulling from etcd
8fdb1fc20e24: Pull complete
436b7dc2bc75: Pull complete
05135444fe12: Pull complete
e462d783cd7f: Pull complete
ecff8ef6851d: Pull complete
Digest: sha256:dd75ec974b0a2a6f6bb47001ba09207976e625db898d1b16735528c009cb171c
Status: Downloaded newer image for registry.k8s.io/etcd:3.5.6-0
registry.k8s.io/etcd:3.5.6-0
```

3.3 Let's **list** the images by executing the below command.

```
# docker image ls
```

Output:

```
[root@docker-engine ~]#docker image ls
REPOSITORY                                TAG       IMAGE ID       CREATED         SIZE
alpine                                    latest    042a816809aa   4 weeks ago     7.05MB
registry.k8s.io/etcd                      3.5.6-0   fce326961ae2   2 months ago    299MB
alpine/git                                latest    22d84a66cda4   2 months ago    43.6MB
quay.io/fluentd_elasticsearch/fluentd     v2.5.2    88d2405af441   3 years ago     140MB
```

3.4 Let's **inspect** the image by executing the below command.

```
# docker image inspect alpine
```

Output:

```
[root@docker-engine ~]# docker image inspect alpine
[
    {
        "Id": "sha256:042a816809aac8d0f7d7cacac7965782ee2ecac3f21bcf9f24b1de1a7387b769",
        "RepoTags": [
            "alpine:latest"
        ],
        "RepoDigests": [
            "alpine@sha256:f271e74b17ced29b915d351685fd4644785c6d1559dd1f2d4189a5e851ef753a"
        ],
        "Parent": "",
        "Comment": "",
        "Created": "2023-01-09T17:05:20.656498283Z",
        "Container": "d4d39cab50d7e505e946044f9131e99602e21f02d0137599e85a70c0b2b7cd15",
        "ContainerConfig": {
            "Hostname": "d4d39cab50d7",
            "Domainname": "",
            "User": "",
            "AttachStdin": false,
            "AttachStdout": false,
            "AttachStderr": false,
            "Tty": false,
            "OpenStdin": false,
            "StdinOnce": false,
            "Env": [
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
            ],
            "Cmd": [
                "/bin/sh",
                "-c",
                "#(nop) ",
                "CMD [\"/bin/sh\"]"
```

4 Cleanup:

4.1 Let's **remove all** the images.

```
# docker image rm `docker image ls -a -q`
```

Output:

```
[root@docker-engine ~]#docker image rm `docker image ls -a -q` -f
Untagged: alpine:latest
Untagged: alpine@sha256:f271e74b17ced29b915d351685fd4644785c6d1559dd1f2d4189a5e851ef753a
Deleted: sha256:042a816809aac8d0f7d7cacac7965782ee2ecac3f21bcf9f24b1de1a7387b769
Deleted: sha256:8e012198eea15b2554b07014081c85fec4967a1b9cc4b65bd9a4bce3ae1c0c88
Untagged: registry.k8s.io/etcd:3.5.6-0
Untagged: registry.k8s.io/etcd@sha256:dd75ec974b0a2a6f6bb47001ba09207976e625db898d1b16735528c009cb171c
Deleted: sha256:fce326961ae2d51a5f726883fd59d2a8c2ccc3e45d3bb859882db58e422e59e7
Deleted: sha256:dd51ec71200a33ec8802733c327a6bafa118e8f2d95621b94598d45613364f47
Deleted: sha256:73e540e294ec705526e6dc1258f1a0f3c6bcbc0eb6a9930c4e7884d90608a975
Deleted: sha256:b80de5608edd46a9560702695bfcfb2232106f54520fb4da3c19400714e029a2
Deleted: sha256:053c48a665f6c39418166d71d7dfaaa51d80020437b8656488e296e4f3a84144
Deleted: sha256:9fce6bd02a21068901a96271d3160c2ce9da9c83c152eeb22312f65226da108c
Untagged: alpine/git:latest
Untagged: alpine/git@sha256:66b210a97bc07bfd4019826bcd13a488b371a6cbe2630a4b37d23275658bd3f2
Deleted: sha256:22d84a66cda4e50afd36fcb8e3b654f7bfe5f7294fae543c97af9df2c10ef858
Deleted: sha256:2e2d14ca6257b9d7789e3497c1b8e53e4923c9ce0a4e5049c6b4291ad8b327e8
Deleted: sha256:4d981ca7b935faafd47b10f31fe065da9207fb34a30760aad7dee44fdb39fee3
Deleted: sha256:e5e13b0c77cbb769548077189c3da2f0a764ceca06af49d8d558e759f5c232bd
Untagged: quay.io/fluentd_elasticsearch/fluentd:v2.5.2
Untagged: quay.io/fluentd_elasticsearch/fluentd@sha256:aec118bb3d1c4af358c1d495b14c12781c4ab5e8cfb455ed
2750e31d
Deleted: sha256:88d2405af441ece5a736e4f2add13b1f3f19d7c298a444c8b34d9ee414bcb3af
Deleted: sha256:598f9ac266b4cd6c24a52cb0f3a6aad76230cadafa5594d7d2bafe7eb23778b0
Deleted: sha256:b66a667c20bd19718b3d5a95b8a6bfc5b1cb979b8c4bd178de68d03fea2cac02
Deleted: sha256:af9ecb6d27bf6d3a3206ec5c4b2f144fa1cbd17758c3b9f460982f891dc0ab08
Deleted: sha256:afa247bb5d9d47559788ce3c9e798bc89c2a68e854aacbec5606ad0f5b330606
Deleted: sha256:5973f2ea7ee5ea188b9c8d575a31ba439f40cab8a0f4c1eb6aa38bbc27948910
Deleted: sha256:9a9d8e2730ea47015da13152e3b066fe7f17be36c4738025566e9c6b1cc46b7f
Deleted: sha256:cf5b3c6798f77b1f78bf4e297b27cfa5b6caa982f04caeb5de7d13c255fd7a1e
```

4.2 Let's **verify** images are removed.

```
# docker image ls
```

Output:

```
[root@docker-engine ~]#docker image ls
REPOSITORY    TAG         IMAGE ID    CREATED    SIZE
```