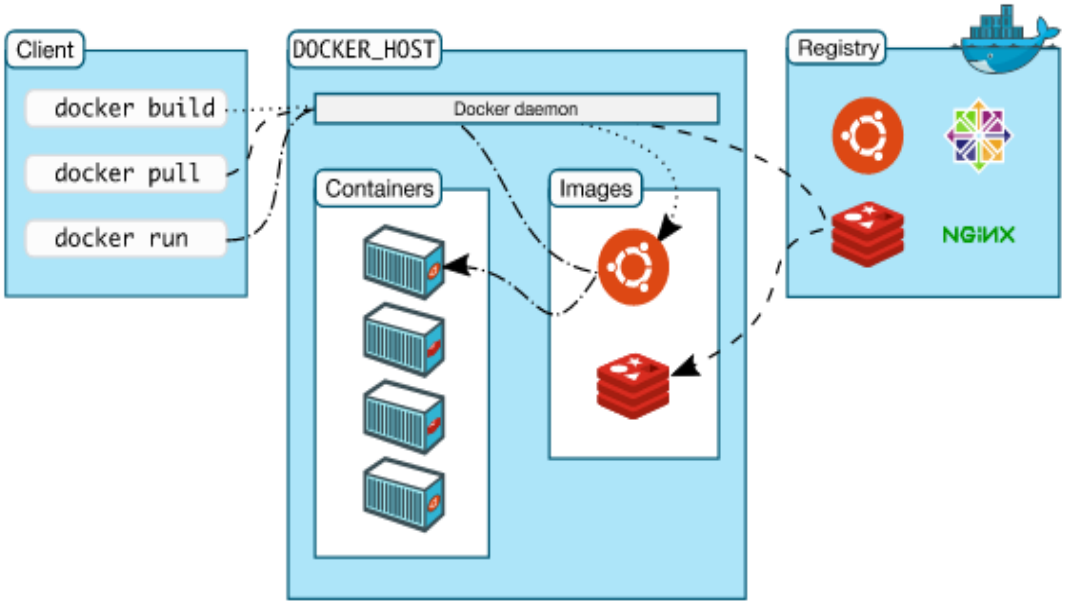
Lab 3 - Managing Container Images

Introduction

In this lab, you will learn how to **search**, **pull** and **remove** container images from the docker registry. We will be creating **Docker Hub** account, to pull and push the images.

1. Search, Pull and Remove container images

In Docker, everything is based on Images. An image is a combination of a file system and parameters.



1.1 Login as **“root”** user on **aio110** host:

Copy

```
ssh root@aio110
```

1.2 Search for a suitable image by using the docker **search** command to find all the images that contain the term **fedor**a.

Copy

```
docker search fedora
```

1.3 Identified a suitable image, **fedora** and now download it using the docker **pull** command.

Copy

```
docker pull fedora
```

Output:

```
Using default tag: latest

latest: Pulling from library/fedora

a8ee583972c2: Pull complete

Digest:
sha256:25f7dac76b2c88d8b7e0b1d6213d3406e77c7f230bfa1e66bd1cbb81a944eaa
f

Status: Downloaded newer image for fedora:latest
```

1.4 List the images locally on host.

Copy

```
docker images
```

Output:

```
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE

fedora              latest              422dc563ca32        3 months
ago        252MB
```

1.5 Inspect the **fedora** image, to display detailed information

Copy

```
docker inspect fedora
```

Output:


```
[

    {

        "Id":
"sha256:422dc563ca3260ad9ef5c47a1c246f5065d7f177ce51f4dd208efd82967ff1
82",

        "RepoTags": [

            "fedora:latest"

        ],

        "RepoDigests": [


"fedora@sha256:25f7dac76b2c88d8b7e0b1d6213d3406e77c7f230bfa1e66bd1cbb8
1a944eaaf"

        ],

        "Parent": "",

        "Comment": "",

        "Created": "2017-11-14T21:07:08.475840838Z",

        "Container":
"8d3b6f1c41196574f7c98869cf8ca90a8f4e5d67a0534753a8df6f3546aaab19",

        "ContainerConfig": {

            "Hostname": "8d3b6f1c4119",

            "Domainname": "",

            "User": "",

            "AttachStdin": false,
```

```
            "AttachStdout": false,
            "AttachStderr": false,
            "Tty": false,
<snip>
```

1.6 Run a container, refer to a tagged image

Copy

```
docker run --name fedora_custom -it fedora /bin/bash
```

Sample Output:

```
[root@f36dfec482af /]#
```

1.7 Installing Maria dB-server packages inside the image

Copy

```
dnf install mariadb-server -y
```

Copy

```
dnf install git -y
```

Note: Fedora uses **dnf** as a Package Manager instead of yum.

1.8 Exit from the interactive mode.

Copy

```
exit
```

1.9 To list all the containers, run the below command.

Copy

```
docker ps -a
```

Output:

```
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS                      PORTS               NAMES

ef321014ebe2        fedora              "/bin/bash"         7 minutes
ago       Exited (0) 3 minutes ago                        fedora_custom
```

1.10 Create a new image from the container on which we made changes by running "**docker commit**" command

Syntax:

```
docker commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]
```

Copy

```
docker commit -m "Added mariadb-server and git" -a "OneCloud"
fedora_custom fedora_custom_image:ver1
```

Sample Output:

```
sha256:71cac0908b698a3783251966b0f2205255a84e1e390ae7976907c9b9a0ce8a1
8
```

Here we used the **docker commit** command. We specified two flags: **-m** and **-a**.

- **-m** flag allows us to specify a *commit message*, much like we would with a commit on a version control system.
- **-a** flag allows us to specify an *author* for our update.

1.11 Check images list.

Copy

```
docker images
```

Output:

```
REPOSITORY                          TAG                 IMAGE ID
CREATED             SIZE

fedora_custom_image                 ver1
71cac0908b69        2 minutes ago       690MB

fedora                              latest
422dc563ca32        3 months ago        252MB
```

2. Creating Docker Hub Account

Docker Hub is a registry service on the cloud that allows you to download Docker images that are built by other communities. You can also upload your own Docker built images to Docker hub. In this chapter, we will see how to download and the use the Jenkins Docker image from Docker hub.

The **official site** for Docker hub is − https://www.docker.com/community-edition#/add_ons

2.1 Copy the below link and **open the web browser**, paste the **URL** and you need to do a simple sign-up on Docker hub

Copy

```
https://hub.docker.com/
```

2.2 Once you have signed up, Verify the your Email id and login to Docker Hub account.

3. Push an image to Docker Hub

3.1 Login into docker hub through CLI

Syntax:

```
docker login -u your-docker-id
```

Note: To run the above command, replace "**your-docker-id**" with your **login_id** .

Save the **Docker Id** as a variable to refer it in the next command:

Copy

```
docker_id=`docker info | grep Username | cut -d " " -f 2`

echo $docker_id
```

Copy

```
docker tag fedora_custom_image:ver1
$docker_id/fedora_custom_image:ver1
```

Copy

```
docker push $docker_id/fedora_custom_image:ver1
```

Sample Output:

```
The push refers to a repository
[docker.io/nagavenikb/fedora_custom_image]

c0e1f7dfd62a: Pushed

d32459d9ce23: Mounted from library/fedora

ver1: digest:
sha256:cd2a46a55f27a44df8ba13fad865f8a8f82482d8c50cd0b8f5c909c80342012
f size: 742
```

Note: Login to Docker Hub account and verify the "**fedora_cutom_image:ver1**" is pushed successfully..

4. Pull an image from Docker Hub

4.1 Verify the list of docker images :

Copy

```
docker images
```

4.2 Let us remove a local copy of "**fedora_cutom_image:ver1**" to pull the same image from Docker Hub which we just pushed:

Copy

```
docker rmi $docker_id/fedora_custom_image:ver1
```

Copy

```
docker images
```

Copy

```
docker pull $docker_id/fedora_custom_image:ver1
```

Copy

```
docker images
```

Now, log out from docker hub account:

Copy

```
docker logout
```

5. Cleanup

5.1 To remove all the containers run the below commands:

Copy

```
docker rm `docker ps -a -q` -f
```

Sample Output:

```
89d2c6c2b29e
```

5.2 To remove all the images run the below commands:

Copy

```
docker rmi `docker images -q` -f
```

Sample Output:

```
Untagged: centos:latest

Untagged:
centos@sha256:2671f7a3eea36ce43609e9fe7435ade83094291055f1c96d9d1d1d7c
0b986a5d

Deleted:
sha256:ff426288ea903fcf8d91aca97460c613348f7a27195606b45f19ae91776ca23
d

Deleted:
sha256:e15afa4858b655f8a5da4c4a41e05b908229f6fab8543434db79207478511ff
7

Untagged: hello-world:latest

Untagged: hello-
world@sha256:083de497cff944f969d8499ab94f07134c50bcf5e6b9559b27182d3fa
80ce3f7

Deleted:
sha256:f2a91732366c0332ccd7afd2a5c4ff2b9af81f549370f7a19acd460f87686bc
7

Deleted:
sha256:f999ae22f308fea973e5a25b57699b5daf6b0f1150ac2a5c2ea9d7fecee50fd
f
```

5.3 Verify that containers are removed:

Copy

```
docker ps
```

Output:

```
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
```

5.4 Verify that docker images are removed:

Copy

```
docker images
```

Output:

```
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
```