Dockerization

Toward an Agile Infrastructure

Ahmed Hassanien
Lead Software Engineer
Gabrianoo Ahmed Hassanien @aka_gabrianoo
Garage Education @GarageEducation
eng.ahmedgaber@gmail.com

April 14, 2020

Table of contents

Introduction - Docker Overview

What is docker?

- Docker is an OS-level virtualization tool.

What is docker?

- Docker is an OS-level virtualization tool.
- Docker is an open platform for developing, shipping, and running applications.

What is docker?

- Docker is an OS-level virtualization tool.
- Docker is an open platform for developing, shipping, and running applications.
- Docker provides tools, and a platform to manage the lifecycle of your containers:

What is docker?

- Docker is an OS-level virtualization tool.
- Docker is an open platform for developing, shipping, and running applications.
- Docker provides tools, and a platform to manage the lifecycle of your containers:
    - Develop your application and its supporting components using containers.

What is docker?

- Docker is an OS-level virtualization tool.
- Docker is an open platform for developing, shipping, and running applications.
- Docker provides tools, and a platform to manage the lifecycle of your containers:
  - Develop your application and its supporting components using containers.
  - The container becomes the unit for distributing and testing your application.

What is docker?

- Docker is an OS-level virtualization tool.
- Docker is an open platform for developing, shipping, and running applications.
- Docker provides tools, and a platform to manage the lifecycle of your containers:
    - Develop your application and its supporting components using containers.
    - The container becomes the unit for distributing and testing your application.
    - When you are ready, deploy your application into your production environment, as a container or an orchestrated service.

What is docker?

- Docker is an OS-level virtualization tool.
- Docker is an open platform for developing, shipping, and running applications.
- Docker provides tools, and a platform to manage the lifecycle of your containers:
  - Develop your application and its supporting components using containers.
  - The container becomes the unit for distributing and testing your application.
  - When you are ready, deploy your application into your production environment, as a container or an orchestrated service.
  - This works the same whether your production environment is a local data center, a cloud provider, or a hybrid of the two.

Docker Architecture

- Docker uses a client-server architecture.

Docker Architecture

- Docker uses a client-server architecture.

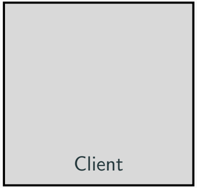

Figure 1: Docker Architecture

Docker Architecture

- Docker uses a client-server architecture.

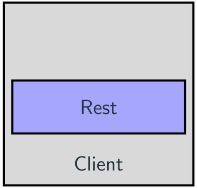

Figure 1: Docker Architecture

Docker Architecture

- Docker uses a client-server architecture.

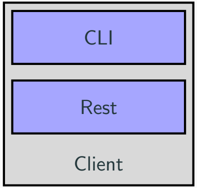

Figure 1: Docker Architecture

Docker Architecture

- Docker uses a client-server architecture.

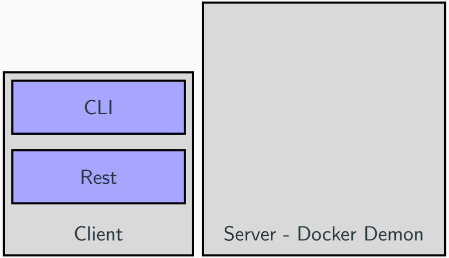

Figure 1: Docker Architecture

Docker Architecture

- Docker uses a client-server architecture.

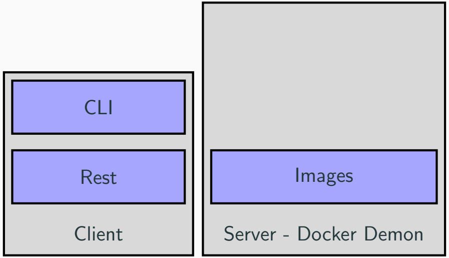

Figure 1: Docker Architecture

Docker Architecture

- Docker uses a client-server architecture.

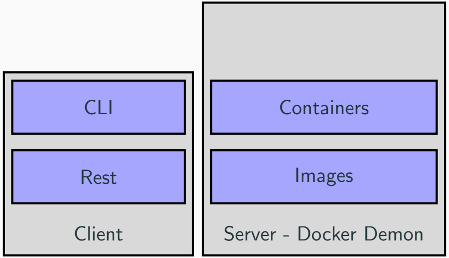

Figure 1: Docker Architecture

Docker Architecture

- Docker uses a client-server architecture.

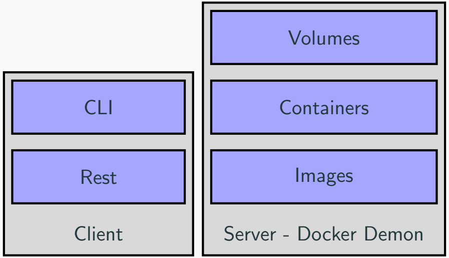

Figure 1: Docker Architecture

Docker Architecture

- Docker uses a client-server architecture.

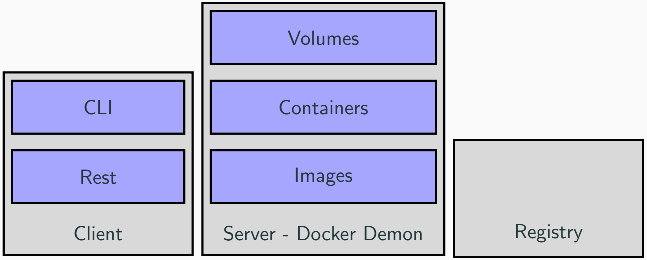

Figure 1: Docker Architecture

Docker Architecture

- Docker uses a client-server architecture.

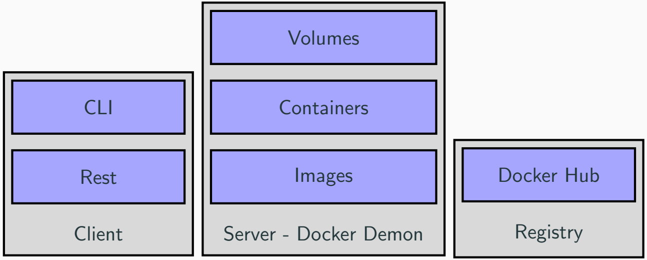

Figure 1: Docker Architecture

Docker Demon (Server)

- The Docker daemon (`dockerd`) listens for Docker API requests and manages Docker objects.

Docker Demon (Server)

- The Docker daemon (`dockerd`) listens for Docker API requests and manages Docker objects.
- A daemon can also communicate with other daemons to manage Docker services.

Docker Objects

- **Images** are a read-only template with instructions for creating a Docker container.

Docker Objects

- **Images** are a read-only template with instructions for creating a Docker container.
- **Containers** are a runnable instance of an image. They live as long as the under laying process is running.

Docker Objects

- **Images** are a read-only template with instructions for creating a Docker container.
- **Containers** are a runnable instance of an image. They live as long as the under laying process is running.
- **Volumes** are the preferred mechanism for persisting data generated by and used by Docker containers.

Docker Client

- The Docker client (`docker`) is the primary way that many Docker users interact with Docker.

Docker Client

- The Docker client (`docker`) is the primary way that many Docker users interact with Docker.
- When you use commands such as (`docker run`), the client sends these commands to (`dockerd`), which carries them out.

Docker Client

- The Docker client (`docker`) is the primary way that many Docker users interact with Docker.
- When you use commands such as (`docker run`), the client sends these commands to (`dockerd`), which carries them out.
- The docker command uses the Docker API and can communicate with one or more docker daemons.

Docker Registry

- A Docker registry stores Docker images.

Docker Registry

- A Docker registry stores Docker images.
- Docker Hub is a public registry that anyone can use, and by default Docker configurations looks for the images on Docker Hub.

Docker Registry

- A Docker registry stores Docker images.
- Docker Hub is a public registry that anyone can use, and by default Docker configurations looks for the images on Docker Hub.
- Docker Hub is not the only registry in the market, and you can use your own docker registry.

Docker in Terminal

docker version

- `version` Show the Docker version information

docker version

- `version` Show the Docker version information

```
docker --version # -v or version
# Docker version 19.03.5, build 633a0ea
```

docker run

- `run` Run a command in a new container

docker run

- run Run a command in a new container

```
docker run hello-world
# Unable to find image hello-world:latest locally
#    latest: Pulling from library/hello-world
#    ca4f61b1923c: Pull complete
#    Digest: sha256:↩
    ca0eeb6fb05351dfc8759c20733c91def84cb8007aa89a5bf606bc8b315b9fc7
#    Status: Downloaded newer image for hello-world:latest
#
#    Hello from Docker!
#    This message shows that your installation appears to be working ↩
    correctly.
```

docker pull

- `run` Pull an image or a repository from a registry

docker pull

- `run` Pull an image or a repository from a registry

```
docker pull docker/whalesay
# Using default tag: latest
# latest: Pulling from docker/whalesay
# Image docker.io/docker/whalesay:latest uses outdated schema1 ↩
    manifest format. Please upgrade to a schema2 image for better ↩
    future compatibility. More information at https://docs.docker.com↩
    /registry/spec/deprecated-schema-v1/
# e190868d63f8: Pull complete
# 909cd34c6fd7: Pull complete
# 0b9bfabab7c1: Pull complete
# a3ed95caeb02: Pull complete
# 00bf65475aba: Pull complete
# c57b6bcc83e3: Pull complete
# 8978f6879e2f: Pull complete
# 8eed3712d2cf: Pull complete
# Digest: sha256:178598↩
    e51a26abbc958b8a2e48825c90bc22e641de3d31e18aaf55f3258ba93b
# Status: Downloaded newer image for docker/whalesay:latest
# docker.io/docker/whalesay:latest
```

docker ps

- ps List containers

docker ps

- ps List containers

```
docker ps
# CONTAINER ID     IMAGE     COMMAND     CREATED     STATUS     PORTS     ↩
    NAMES
docker ps -a # --all
# CONTAINER ID     IMAGE        COMMAND            CREATED          STATUS↩
                          NAMES
# 5bf7bfb3de4a     whalesay     cowsay Garage     8 minutes ago     Exited↩
     (0) 3 minutes ago     friendly_roentgen
docker ps -a -q
# 44b30241b056
# 6a5acfb94078
# eefc7f9cf8a6
# b361e061f108
# 5bf7bfb3de4a
```

docker start

- `start` Start one or more stopped containers

docker start

- start Start one or more stopped containers

```
docker start -a -i friendly_roentgen
#  __________________
# < Garage Education >
#  ------------------
#     \
#      \
#       \
#                     ##        .
#               ## ## ##       ==
#            ## ## ## ##      ===
#        /""""""""""""""""___/ ===
#   ~~~ {~~ ~~~~ ~~~ ~~~~ ~~ ~ /  ===- ~~~
#        \______ o          __/
#          \    \        __/
#           \____\______/
```

docker rm

- `rm` Remove one or more containers

docker rm

- `rm` Remove one or more containers

```
docker rm 44b30241b056
# 44b30241b056
docker rm $(docker ps -a -q)
# 6a5acfb94078
# eefc7f9cf8a6
# b361e061f108
# 5bf7bfb3de4a
docker ps -a
# CONTAINER ID    IMAGE    COMMAND    CREATED    STATUS    NAMES
docker run --rm docker/whalesay cowsay Garage Education
docker ps -a
# CONTAINER ID    IMAGE    COMMAND    CREATED    STATUS    NAMES
```

docker rename

- `rename` Rename a container

docker rename

- `rename` Rename a container

```
docker run docker/whalesay cowsay Garage Education
#  __________________
# < Garage Education >
#  ------------------
#     \
#      \
#       \
#                     ##        .
#               ## ## ##       ==
#            ## ## ## ##      ===
#        /""""""""""""""""___/ ===
#   ~~~ {~~ ~~~~ ~~~ ~~~~ ~~ ~ /  ===- ~~~
#        \______ o          __/
#         \    \        __/
#           \____\______/
docker ps -a
# CONTAINER ID        IMAGE        COMMAND          CREATED ↩
                STATUS                      NAMES
# 6de4ea6d7231        whalesay     cowsay Garage   3 seconds ago ↩
          Exited (0) 2 seconds ago    loving_antonelli
docker rename loving_antonelli whalesay
docker ps -a
# CONTAINER ID        IMAGE        COMMAND          CREATED
```

docker images

- `images` List images

docker images

- `images` List images

```
docker images
# REPOSITORY          TAG                 IMAGE ID            CREATED ↩
                SIZE
# ubuntu              latest              4e5021d210f6        3 weeks ↩
    ago         64.2MB
# docker/whalesay     latest              6b362a9f73eb        4 years ↩
    ago         247MB
```

docker rmi

- `rmi` Remove one or more images

docker rmi

- `rmi` Remove one or more images

```
docker rmi ubuntu
# Untagged: ubuntu:latest
# Untagged: ubuntu@sha256:↩
    bec5a2727be7fff3d308193cfde3491f8fba1a2ba392b7546b43a051853a341d
# Deleted: sha256:4↩
    e5021d210f65ebe915670c7089120120bc0a303b90208592851708c1b8c04bd
# Deleted: sha256:1↩
    d9112746e9d86157c23e426ce87cc2d7bced0ba2ec8ddbdfbcc3093e0769472
# Deleted: sha256:↩
    efcf4a93c18b5d01aa8e10a2e3b7e2b2eef0378336456d8653e2d123d6232c1e
# Deleted: sha256:1↩
    e1aa31289fdca521c403edd6b37317bf0a349a941c7f19b6d9d311f59347502
# Deleted: sha256:↩
    c8be1b8f4d60d99c281fc2db75e0f56df42a83ad2f0b091621ce19357e19d853
docker rmi $(docker images -q)
```

docker search

- `search` Search the Docker Hub for images

docker search

- `search` Search the Docker Hub for images

```
docker search ubuntu
# NAME                              DESCRIPTION ↩
                                         STARS  OFFICIAL  AUTOMATED
# ubuntu                            Ubuntu is a Debian-based Linux ↩
     operating sys     10749  [OK]
# dorowu/ubuntu-desktop-lxde-vnc    Docker image to provide HTML5 VNC ↩
     interface      414              [OK]
# rastasheep/ubuntu-sshd            Dockerized SSH service, built on ↩
     top of offi     245              [OK]
# consol/ubuntu-xfce-vnc            Ubuntu container with "headless" ↩
     VNC session     214              [OK]
# ubuntu-upstart                    Upstart is an event-based ↩
     replacement for th     107    [OK]
# ansible/ubuntu14.04-ansible       Ubuntu 14.04 LTS with ansible ↩
                       98               [OK]
# neurodebian                       NeuroDebian provides neuroscience ↩
     research s     68     [OK]
# ubuntu-debootstrap                debootstrap --variant=minbase --↩
     components=m     44     [OK]
```

docker login

- `login` Log in to a Docker registry

docker login

- `login` Log in to a Docker registry

```
docker login
# Login with your Docker ID to push and pull images from Docker Hub. ↩
    If you do not have a Docker ID, head over to https://hub.docker.↩
    com to create one.
# Username: ahmedhassanien
# Password:
# Login Succeeded
```

docker logout

- `logout` Log out from a Docker registry

docker logout

- logout Log out from a Docker registry

```
docker logout
# Removing login credentials for https://index.docker.io/v1/
```

index.docker.io

- `index.docker.io` is hosted on AWS :-)

index.docker.io

- index.docker.io is hosted on AWS :-)

```
nslookup index.docker.io
# Server:   192.168.1.22
# Address:  192.168.1.22#53

# Non-authoritative answer:
# index.docker.io  canonical name = elb-io.us-east-1.aws.dckr.io.
# elb-io.us-east-1.aws.dckr.io  canonical name = us-east-1-elbio-↩
    rm5bon1qaeo4-623296237.us-east-1.elb.amazonaws.com.
# Name:  us-east-1-elbio-rm5bon1qaeo4-623296237.us-east-1.elb.↩
    amazonaws.com
# Address: 52.3.104.67
# Name:  us-east-1-elbio-rm5bon1qaeo4-623296237.us-east-1.elb.↩
    amazonaws.com
# Address: 3.220.75.233
# Name:  us-east-1-elbio-rm5bon1qaeo4-623296237.us-east-1.elb.↩
    amazonaws.com
# Address: 34.193.164.221
```