

Dockerization

Toward an Agile Infrastructure

Ahmed Hassanien

Lead Software Engineer

 Gabrianoo  Ahmed Hassanien  @aka_gabrianoo

 Garage Education  @GarageEducation

 eng.ahmedgaber@gmail.com

April 29, 2020

Table of contents

1. Introduction - Docker Overview
2. Docker - Short-lived containers
3. Docker - Long-lived containers
4. Docker - More than one container app

Docker - Long-lived containers

docker detached

- `-d`, `--detach` Run container in background and print container ID.

```
docker run --rm httpd
# AH00558: httpd: Could not reliably determine the servers fully ↵
  qualified domain name, using 172.17.0.2. Set the ServerName ↵
  directive globally to suppress this message
# AH00558: httpd: Could not reliably determine the servers fully ↵
  qualified domain name, using 172.17.0.2. Set the ServerName ↵
  directive globally to suppress this message
# [Sat Apr 18 01:31:07.312598 2020] [mpm_event:notice] [pid 1:tid ↵
  140057600050304] AH00489: Apache/2.4.43 (Unix) configured -- ↵
  resuming normal operations
# [Sat Apr 18 01:31:07.312772 2020] [core:notice] [pid 1:tid ↵
  140057600050304] AH00094: Command line: httpd -D FOREGROUND
docker run --rm -d httpd
```

docker expose container port

- `port` List port mappings or a specific mapping for the container.
- `-p`, `--publish list` Publish a container's port(s) to the host.

```
docker run -d httpd
docker run -d httpd
docker run -d httpd
docker run --rm -d httpd
docker run --rm -d -p 80:80 httpd
docker run --rm -d -p 8080:80 httpd
docker port nervous_lamarr
# 80/tcp -> 0.0.0.0:8080
```

docker bridge networking

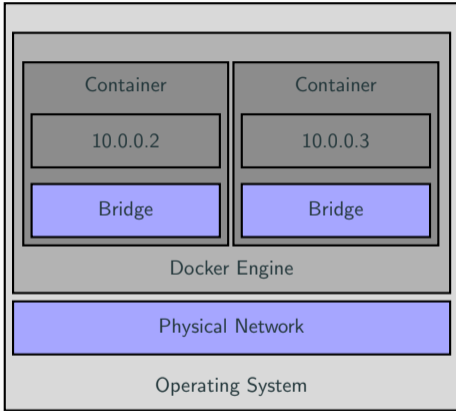


Figure 1: Docker Bridge Networking

docker show container logs

- logs Fetch the logs of a container.
- -t, --timestamps Show timestamps.
- -f, --follow Follow log output.

```
docker logs nervous_lamarr
# AH00558: httpd: Could not reliably determine the servers fully ↵
  qualified domain name, using 172.17.0.6. Set the ServerName ↵
  directive globally to suppress this message
# AH00558: httpd: Could not reliably determine the servers fully ↵
  qualified domain name, using 172.17.0.6. Set the ServerName ↵
  directive globally to suppress this message
# [Sat Apr 18 18:02:11.771692 2020] [mpm_event:notice] [pid 1:tid ↵
  139892595709056] AH00489: Apache/2.4.43 (Unix) configured -- ↵
  resuming normal operations
# [Sat Apr 18 18:02:11.772107 2020] [core:notice] [pid 1:tid ↵
  139892595709056] AH00094: Command line: httpd -D FOREGROUND
# 172.17.0.1 - - [18/Apr/2020:18:03:02 +0000] "GET / HTTP/1.1" 200 45
# 172.17.0.1 - - [18/Apr/2020:18:03:02 +0000] "GET /favicon.ico HTTP↵
  /1.1" 404 196
docker logs -t nervous_lamarr
docker logs -f nervous_lamarr
```

docker restart

- restart Restart one or more containers.
- -t, --time int Seconds to wait for stop before killing the container (default 10).

```
docker restart nervous_lamarr
# nervous_lamarr
docker ps -a
# CONTAINER ID        IMAGE          COMMAND                  CREATED             STATUS              PORTS                    NAMES
# 89d48b6635c3        httpd         "httpd-foreground"     36 minutes ago    Up 3↔
# 357e2aafd0d3        httpd         "httpd-foreground"     40 minutes ago    Up ↔
# 40 minutes         0.0.0.0:80->80/tcp    gallant_lewin
docker restart -t 5 nervous_lamarr
# nervous_lamarr
```


docker stop container

- stop Stop one or more running containers.
- -t, --time int Seconds to wait for stop before killing it (default 10).

```
docker stop cranky_chatterjee
# cranky_chatterjee
docker ps -a
# CONTAINER ID      IMAGE      COMMAND                  CREATED          ↵
  STATUS           NAMES
# d16271db5d73     httpd     "httpd-foreground"     17 minutes ago  ↵
  Exited (0) 3 seconds ago    cranky_chatterjee
docker rm cranky_chatterjee
# cranky_chatterjee
docker stop -t 5 cool_spence
# cool_spence
docker rm cool_spence
# cool_spence
```

docker kill container

- Kill Kill one or more running containers.

```
docker kill gracious_jang
# gracious_jang
docker rm gracious_jang
# gracious_jang
```

docker stats

- stats Display a live stream of container(s) resource usage statistics.
- -a, --all Show all containers (default shows just running).
- --no-stream Disable streaming stats and only pull the first result.

```
docker stats nervous_lamarr
# CONTAINER ID   NAME           CPU %       MEM USAGE / LIMIT   MEM↔
%       NET I/O       BLOCK I/O   PIDS
# 89d48b6635c3   nervous_lamarr 0.01%      6.109MiB / 1.943GiB ←
0.31%    9.29kB / 4.54kB  0B / 0B    82
docker stats -a
# CONTAINER ID   NAME           CPU %       MEM USAGE / LIMIT   MEM↔
%       NET I/O       BLOCK I/O   PIDS
# 89d48b6635c3   nervous_lamarr 0.01%      6.109MiB / 1.943GiB ←
0.31%    9.29kB / 4.54kB  0B / 0B    82
# 357e2aafd0d3   gallant_lewin  0.02%      5.996MiB / 1.943GiB ←
0.30%    2.01kB / 558B   0B / 0B    82
docker stats -a --no-stream
```

- top Display the running processes of a container.

```
docker top nervous_lamarr
# PID      USER      TIME      COMMAND
# 3634     root      0:00      httpd -DFOREGROUND
# 3668     bin       0:00      httpd -DFOREGROUND
# 3669     bin       0:00      httpd -DFOREGROUND
# 3670     bin       0:00      httpd -DFOREGROUND
```

docker pause and unpause

- pause Pause all processes within one or more containers.
- unpause Unpause all processes within one or more containers.

```
docker pause nervous_lamarr
# nervous_lamarr
docker ps -a
# CONTAINER ID        IMAGE          COMMAND                  CREATED             STATUS              PORTS              NAMES
# 89d48b6635c3        httpd         httpd-foreground        36 minutes ago     Up 36              0.0.0.0:8080->80/tcp nervous_lamarr
# 357e2aafd0d3        httpd         httpd-foreground        40 minutes ago     Up 40              0.0.0.0:80->80/tcp  gallant_lewin
docker unpause nervous_lamarr
# nervous_lamarr
docker ps -a
# CONTAINER ID        IMAGE          COMMAND                  CREATED             STATUS              PORTS              NAMES
# 89d48b6635c3        httpd         httpd-foreground        36 minutes ago     Up 36              0.0.0.0:8080->80/tcp nervous_lamarr
# 357e2aafd0d3        httpd         httpd-foreground        40 minutes ago     Up 40              0.0.0.0:80->80/tcp  gallant_lewin
```

docker exec

- `exec` Run a command in a running container.
- `-i`, `--interactive` Keep STDIN open even if not attached.
- `-t`, `--tty` Allocate a pseudo-TTY.

```
docker exec nervous_lamarr /bin/ls
docker exec nervous_lamarr /bin/bash
docker ps -a
# CONTAINER ID        IMAGE          COMMAND                  CREATED              STATUS              PORTS              NAMES
# 89d48b6635c3       httpd         httpd-foreground        36 minutes ago      Up 36               0.0.0.0:8080->80/tcp nervous_lamarr
docker exec -i nervous_lamarr /bin/bash
docker ps -a
# CONTAINER ID        IMAGE          COMMAND                  CREATED              STATUS              PORTS              NAMES
# 89d48b6635c3       httpd         "httpd-foreground"      36 minutes ago      Up                 0.0.0.0:8080->80/tcp nervous_lamarr
docker exec -i -t nervous_lamarr /bin/bash
# root@89d48b6635c3:/usr/local/apache2# cat htdocs/index.html
# <html><body><h1>It works!</h1></body></html>
```

- cp Copy files/folders between a container and the local filesystem.

```
docker cp nervous_lamarr:/usr/local/apache2/htdocs/index.html .  
vim index.html  
docker cp index.html nervous_lamarr:/usr/local/apache2/htdocs/index.html
```

- `wait` Block until one or more containers stop, then print their exit codes.

```
docker wait nervous_lamarr
# 0
docker stop nervous_lamarr
# nervous_lamarr
```


docker attach

- attach Attach local standard input, output, and error streams to a running container.

```
docker attach gallant_lewin

docker run --name mysql -e MYSQL_ROOT_PASSWORD=P@ssw0rd -d mysql:5.7
docker run --name mysql -p 3306:3306 -e MYSQL_ROOT_PASSWORD=P@ssw0rd \
-d mysql:5.7
hostname
# hostname.lan
docker run -it --rm mysql:5.7 mysql -hhostname.lan -uroot -pP@ssw0rd
docker run -it --rm mysql:5.7 mysql -h"${hostname}" -uroot -pP@ssw0rd

docker network create wp-network
docker network connect wp-network mysql
docker network inspect wp-network
docker run --name wordpress --network wp-network -p 8080:80 \
-e WORDPRESS_DB_HOST=mysql -e WORDPRESS_DB_NAME=wp_db \
-e WORDPRESS_DB_USER=root -e WORDPRESS_DB_PASSWORD=P@ssw0rd \
-d wordpress
docker network inspect wp-network
docker network rm wp-network
```