

Section 5 - Containers lifecycle

2 - Monitor & Inspect Containers

Objective

The objective is to understand how to monitor the running containers by using:

- The `docker top` command to display the running processes of a container.
- The `docker stats` command to display a live stream (real time view) of container(s) resource usage statistics such as cpu%, memory, disk I/O and network I/O.
- The `docker container inspect` command to display detailed information of a container such as meta-data, configuration etc.

Commands overview

`docker container top` => process list in one container

`docker container inspect` => details of one container config

`docker container stats` => performance stats for all containers

Example (1)

- In this example we will start two containers `nginx` and `mysql` in detach mode:

```
# docker container run -d --name nginx nginx  
# docker container run -d --name mysql -e MYSQL_RANDOM_ROOT_PASSWORD=tr
```

Information about the environment variables that can be used with the mysql Docker image is available at the docker-hub registry:
https://hub.docker.com/_/mysql

Example (2)

- Verify that the "nginx" and "mysql" containers are running:

```
# docker container ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
569f878a0712	mysql	"docker-entrypoint.s..."	21 seconds ago	Up 20 s
082912c9a054	nginx	"nginx -g 'daemon of..."	22 seconds ago	Up 21 s

docker container top

- Use the `docker container top` command to display the running processes of each container:

```
# docker container top mysql
UID          PID        PPID        C    STIME     TTY      TIME          CMD
polkitd      15775      15761       1    06:56     ?        00:00:00      mysqld

# docker container top nginx
UID    PID    PPID    C    STIME     TTY      TIME          CMD
root   15718   15704    0    06:56     ?        00:00:00      nginx: master process nginx -c
101    15752   15718    0    06:56     ?        00:00:00      nginx: worker process
```

Notes:

- We can see that there are 2 processes running inside the nginx container.
- With nginx, there's actually a master process and then it spawns worker processes based on the configuration.
- You can see that a container is more like a process (or more) running isolated in the host OS.

docker container inspect

- Use the `docker container inspect` command to display detailed information of a container such as meta-data, configuration etc...

```
# docker container inspect nginx
[
  {
    "Id": "082912c9a0545c88ca60772f9b757f0173e6a53b674227892b3b64d2287!",
    "Created": "2019-04-09T03:56:57.415293403Z",
    "Path": "nginx",
    "Args": [
      "-g",
      "daemon off;"
    ],
    ...
  ]
]
```

The output of this command is a JSON array containing detailed information about the container.

docker stats

- Use the `docker stats` command to display a live stream (real time view) of container(s) resource usage statistics. The command supports *CPU*, *memory usage*, *memory limit*, and *network IO* metrics.

```
# docker stats
CONTAINER ID      NAME      CPU %      MEM USAGE / LIMIT      MEM %      NET I/O
569f878a0712      mysql     1.04%      374.4MiB / 3.701GiB      9.88%      729B / 0B
082912c9a054      nginx     0.00%      1.359MiB / 3.701GiB      0.04%      729B / 0B
```

The output of the `docker stats` command is a live stream (real time view). To exit from the real time view of the statistics press `<Ctrl-C>`.

docker stats [container name|ID]

- We can specify the one or more container names (or ID) with the "docker stats" to limit the output to the specified container.
- Example

```
# docker stats nginx
CONTAINER ID   NAME      CPU %     MEM USAGE / LIMIT     MEM %    NET I/O     BLOCK
082912c9a054   nginx    0.00%    1.359MiB / 3.701GiB    0.04%    729B / 0B    737kB
```

Note:

These commands are not very helpful on a production environment, where there is a more complicated configuration with many containers running and, in case of cluster, more than one server to monitor. For production environment, tools such as **Prometheus** and **ELK** are used for monitoring and logging*.

Commands Summary

```
docker container run -d --name nginx nginx
docker container run -d --name mysql -e MYSQL_RANDOM_ROOT_PASSWORD=true mys
docker container ls
docker container top mysql
docker container top nginx
docker container inspect mysql
docker container stats --help
docker container stats
docker container ls
```

Exercise

- Ref
- D_S5_L2_Monitor_Containers_ex.md