Dashboard
Assessments
Premium Bootcamps
WeCloud Open
Webinar & Events
Career Paths
Collapse

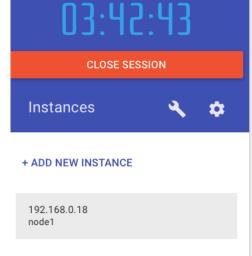
Data Engineer Bootcamp (Full-Time)

HM
HIBAHMOHAMMED O SINDI
haboba1417@hotmail.com
Programs Settings
Sign Out
Notes

Demonstrating Hello World Example | dockerlabs

Tested Infrastructure

Running Hello World Example





Explanation

This image is a prime example of using the scratch image effectively. See hello.c in https://github.com/docker-library/hello-world for the source code of the hello binary included in this image.

So what's happened here? We've called the docker run command, which is responsible for launching containers.

The argument hello-world is the name of the image someone created on dockerhub for us. It will first search for "hello-world" image locally and then search in Dockerhub.

Once the image has been downloaded, Docker turns the image into a running container and executes it.

Did you Know?

1. The Hello World Docker Image is only 1.84 KB size.

```
[node1] (local) root@192.168.0.18 ~
$ docker images
REPOSITORY
                    TAG
                                         IMAGE ID
                                                             CREATED
                                                                                  SIZE
                                                                                  1.84kB
hello-world
                    latest
                                         4ab4c602aa5e
                                                             6 weeks ago
```

1. While running docker ps command, it doesn't display any running container. Reason - It gets executed once and exit immediately.

```
$ docker ps
CONTAINER ID
                                         COMMAND
                                                             CREATED
                                                                                  STATUS
      PORTS
                          NAMES
```

1. You can use docker inspect <imagename> command to inspect about this particular Docker Image.

```
docker inspect 4ab
        "Id": "sha256:4ab4c602aa5eed5528a6620ff18a1dc4faef0e1ab3a5eddeddb410714478c67f",
        "RepoTags": [
              "hello-world:latest"
        ],
"RepoDigests": [
              "hello-world@sha256:0add3ace90ecb4adbf7777e9aacf18357296e799f81cabc9fde470971e499788"
       ],
"Parent":
        "Parent": "",
"Comment": "",
"Created": "2018-09-07T19:25:39.809797627Z",
"Container": "15c5544a385127276a51553acb81ed24a9429f9f61d6844db1fa34f46348e420",
        "ContainerConfig": {
    "Hostname": "15c5544a3851",
              "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 [\"/hello\"]"
             ],
"ArgsEscaped": true,
"Image": "sha256:9a5813f1116c2426ead0a44bbec252bfc5c3d445402cc1442ce9194fc1397027",
"Volumes": null,
"WorkingDir": "",
              "Entrypoint": null,
              "OnBuild": null,
              "Labels": {}
       },
"DockerVersion": "17.06.2-ce",
"Author": "",
"Config": {
    ""instrame": "".
              "Hostname": ""
             "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": [
"/hello"
             ],
"ArgsEscaped": true,
"Image": "sha256:9a5813f1116c2426ead0a44bbec252bfc5c3d445402cc1442ce9194fc1397027",
"Volumes": null,
"WorkingDir": "",
              "Entrypoint": null,
              "OnBuild": null,
              "Labels": null
        },
"Architecture": "amd64",
        "Os": "linux",
"Size": 1840,
        "VirtualSize": 1840,
```

```
"GraphDriver": {
              .
"Data": {
                  "MergedDir": "/var/lib/docker/overlay2/e494ae30abc49ad403ef5c2a32bcb894629ea4da6d4d226fbca70d27ed9a74d8/merged",
"UpperDir": "/var/lib/docker/overlay2/e494ae30abc49ad403ef5c2a32bcb894629ea4da6d4d226fbca70d27ed9a74d8/diff",
"WorkDir": "/var/lib/docker/overlay2/e494ae30abc49ad403ef5c2a32bcb894629ea4da6d4d226fbca70d27ed9a74d8/work"
             },
"Name": "overlay2"
        },
"Metadata": {
    "'actTagT:
              "LastTagTime": "0001-01-01T00:00:00Z"
    }
]
Course Content
Enter code
7
All
Lecture
Recordings
Practices
Chapter
Program Information
Chapter
Surveys
Chapter
Week 00 (Virtual)- Program Preparation
Chapter
Week 01 - SQL
>
Chapter
Week 02 - Python
Chapter
Week 03 - Client Project
Chapter
Week 04 - Linux and AWS
Chapter
Week 05 - Docker and Client Project phase 2
Chapter overview
Sunday - Docker I
[Lecture Material] Docker
[Lab] Software Installation: Docker
[Lab] Account Creation Create your Dockerhub account
[Lab] Workshop Demonstrating Hello World Example
[Lab] Workshop: Work with Docker Image
[Lab] Exercise: Basic Docker Commands
[Lecture Video] Docker Sunday
Monday - Docker II
[Lab] Workshop: Install Zepplin with Docker
[Lab] Workshop: Docker Compose --Flask
[Quiz] Docker Commands Quiz
Tuesday - Real Client Project Phase 2
Wednesday - Real Client Project Phase 2
Thursday - Real Client Project Phase 2
RCP project Submission (Competition)
[Lab] Workshop Demonstrating Hello World Example
```