

Docker Project

Installed Docker on Amazon Linux 2 EC2 Instance by using the following documentation:

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/create-container-image.html>

```
[ec2-user@ip-192-0-0-11 ~]$ docker info
Client:
 Context:    default
 Debug Mode: false
 Plugins:
  buildx: Docker Buildx (Docker Inc., v0.0.0+unknown)

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 1
 Server Version: 20.10.25
 Storage Driver: overlay2
  Backing Filesystem: xfs
  Supports d_type: true
  Native Overlay Diff: true
  userxattr: false
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Cgroup Version: 1
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
 Swarm: inactive
```

Test by running hello-world:

```
[ec2-user@ip-192-0-0-11 ~]$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:88ec0acaa3ec199d3b7eaf73588f4518c25f9d34f58ce9a0df68429c5af48e8d
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

[ec2-user@ip-192-0-0-11 ~]$
```

```
[ec2-user@ip-192-0-0-11 ~]$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
[ec2-user@ip-192-0-0-11 ~]$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED      STATUS      PORTS      NAMES
1741c4e80ef6   hello-world  "/hello"   3 minutes ago  Exited (0)  3 minutes ago  tive_zhukovsky
[ec2-user@ip-192-0-0-11 ~]$
```

Containerizing the apache web application -

Writing the Dockerfile in VSCode:

```
my-docker-repo > Dockerfile > ...
1  # Dockerfile to containerize an apache web application
2  # Base image
3  FROM centos:7
4
5  # Description
6  LABEL description="Dockerfile to containerize an apache web application"
7  # update all packages
8  RUN yum -y update
9
10 # install apache
11 RUN yum install -y httpd
12
13 #Copy web app from local to the container
14 COPY index.html /var/www/html/
15
16 # port on which the app should listen on
17 EXPOSE 80
18
19 # systemctl start httpd
20 # CMD or ENTRYPOINT
21 ENTRYPOINT [ "/usr/sbin/httpd" ]
22
23 # Run the container in the background
24 CMD ["-D", "FOREGROUND"]
```

Pasting the Dockerfile into the docker server:

```
# Dockerfile to containerize an apache web application
# Base image
FROM centos:7

# Description
LABEL description="Dockerfile to containerize an apache web application"
# update all packages
RUN yum -y update

# install apache
RUN yum install -y httpd

#Copy web app from local to the container
COPY index.html /var/www/html/

# port on which the app should listen on
EXPOSE 80

# systemctl start httpd
# CMD or ENTRYPOINT
ENTRYPOINT [ "/usr/sbin/httpd" ]

# Run the container in the background
CMD [ "-D", "FOREGROUND" ]
~
~
~
~
```

Adding the index.html file on the docker server:

```
<html><body><h1 style="color:rgb(224, 64, 64);">WELCOME TO OLGA'S WEBAPP RUNNING INSIDE
A CONTAINER!</h1></body></html>
~
~
~
```

Building the docker image:

```
sh-4.2$ sudo su - ec2-user
Last login: Mon Nov 13 21:26:46 UTC 2023 on pts/0
[ec2-user@ip-192-0-0-11 ~]$ ls -l
total 0
[ec2-user@ip-192-0-0-11 ~]$ vim Dockerfile
[ec2-user@ip-192-0-0-11 ~]$ vim index.html
[ec2-user@ip-192-0-0-11 ~]$ vim index.html
[ec2-user@ip-192-0-0-11 ~]$ ls -l
total 8
-rw-rw-r-- 1 ec2-user ec2-user 392 Nov 15 16:33 Dockerfile
-rw-rw-r-- 1 ec2-user ec2-user 120 Nov 15 16:44 index.html
[ec2-user@ip-192-0-0-11 ~]$ docker build -t olga_apache:v1 .
```

```
Installing : mailcap-2.1.41-2.el7.noarch                    5/6
Installing : httpd-2.4.6-99.el7.centos.1.x86_64           6/6
Verifying  : httpd-tools-2.4.6-99.el7.centos.1.x86_64     1/6
Verifying  : mailcap-2.1.41-2.el7.noarch                   2/6
Verifying  : apr-1.4.8-7.el7.x86_64                       3/6
Verifying  : httpd-2.4.6-99.el7.centos.1.x86_64           4/6
Verifying  : apr-util-1.5.2-6.el7_9.1.x86_64              5/6
Verifying  : centos-logos-70.0.6-3.el7.centos.noarch       6/6

Installed:
  httpd.x86_64 0:2.4.6-99.el7.centos.1

Dependency Installed:
  apr.x86_64 0:1.4.8-7.el7
  apr-util.x86_64 0:1.5.2-6.el7_9.1
  centos-logos.noarch 0:70.0.6-3.el7.centos
  httpd-tools.x86_64 0:2.4.6-99.el7.centos.1
  mailcap.noarch 0:2.1.41-2.el7

Complete!
Removing intermediate container c639759e4657
---> b3290beac0b5
Step 5/7 : EXPOSE 80
---> Running in lae86b0f9e00
Removing intermediate container lae86b0f9e00
---> f3169e827159
Step 6/7 : ENTRYPOINT [ "/usr/sbin/httpd" ]
---> Running in 92c82d725727
Removing intermediate container 92c82d725727
---> 4308bb234ad8
Step 7/7 : CMD ["-D", "FOREGROUND"]
---> Running in be411d96a5de
Removing intermediate container be411d96a5de
---> 35bc01d9cdfd
Successfully built 35bc01d9cdfd
Successfully tagged olga_apache:v1
[ec2-user@ip-192-0-0-11 ~]$
```

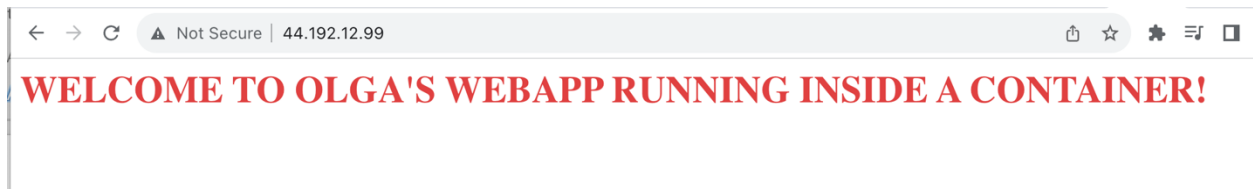
```
[ec2-user@ip-192-0-0-11 ~]$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
olga_apache    v1        35bc01d9cdfd   11 minutes ago 833MB
hello-world    latest    9c7a54a9a43c   6 months ago  13.3kB
centos         7         eeb6ee3f44bd   2 years ago   204MB
[ec2-user@ip-192-0-0-11 ~]$
```

Running the docker container from the docker image:

```
[ec2-user@ip-192-0-0-11 ~]$ docker run -t -p 80:80 olga_apache:v1
```

```
[ec2-user@ip-192-0-0-11 ~]$ docker run -t -p 80:80 olga_apache:v1
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
█
```

Testing the containerized application in the browser:



For the container to run in the background in detached mode:

```
[ec2-user@ip-192-0-0-11 ~]$ docker run -t -d -p 80:80 olga_apache:v1
0bb85e9d78b76f419a9871a69fb3e280b4a683ddd9e45f96efc99ad7e155c2ca
[ec2-user@ip-192-0-0-11 ~]$ █
```

Publishing the docker image to DockerHub:


1. First logging into DockerHub using the docker login command
2. Then tagging the image with the docker tag command
3. Then pushing to DockerHub with the docker push command

```
[ec2-user@ip-192-0-0-11 ~]$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you d
docker.com to create one.
Username: omyro
Password:
WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-sto

Login Succeeded
[ec2-user@ip-192-0-0-11 ~]$ docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
olga_apache     v1          63607d76e49a  18 minutes ago 833MB
hello-world     latest      9c7a54a9a43c  6 months ago  13.3kB
centos          7          eeb6ee3f44bd  2 years ago   204MB
[ec2-user@ip-192-0-0-11 ~]$ docker tag 63607d76e49a omyro/olga_apache:v1
[ec2-user@ip-192-0-0-11 ~]$ docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
olga_apache     v1          63607d76e49a  19 minutes ago 833MB
omyro/olga_apache v1          63607d76e49a  19 minutes ago 833MB
hello-world     latest      9c7a54a9a43c  6 months ago  13.3kB
centos          7          eeb6ee3f44bd  2 years ago   204MB
[ec2-user@ip-192-0-0-11 ~]$
```

```
[ec2-user@ip-192-0-0-11 ~]$ docker push omyro/olga_apache:v1
The push refers to repository [docker.io/omyro/olga_apache]
27eac33c731d: Pushed
183b031595ba: Pushing [=====>] 44.51MB/261.6MB
b9d89790bf5f: Pushing [=====>] 43.08MB/367.7MB
174f56854903: Mounted from library/centos
```

Checking that the image has been successfully published to Docker Hub:




Search Docker Hub

🔍

ExploreRepositoriesOrganizationsHelp

Upgrade

omyro

omyro

Search by repository name

All Content

Create repository

omyro / olga_apache

Contains: Image | Last pushed: a few seconds ago

Inactive

0

0

Public

omyro / Repositories / olga_apache / General

Using 0 of 1 private repositories. [Get more](#)

General


Tags

Builds

Collaborators


Webhooks

Settings

 Add a short description for this repository

The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.

Update

 omyro / olga_apache

Description

This repository does not have a description

Last pushed: a minute ago

Docker commands

To push a new tag to this repository:

docker push omyro/olga_apache:tagname

Public View

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
v1		Image	---	a minute ago

[See all](#)[Go to Advanced Image Management](#)

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds](#)

Upgrade