

Experiment 5

AIM: Working with Dockerfile to Build and Push Docker Image

Steps to Complete:

1. Create following Dockerfile

```
FROM alpine
MAINTAINER SUDIPT
RUN apk update
RUN apk add nodejs
RUN mkdir /app
COPY index.js /app
WORKDIR /app
RUN node index.js
```

```
anuvagarg@Anuvag-MacBook-Air Docker % docker build -f Dockerfile .
[+] Building 19.0s (10/12)
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 174B                                              0.0s
=> [internal] load .dockerignore                                                 0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/alpine:latest                 18.8s
=> [auth] library/alpine:pull token for registry-1.docker.io                   0.0s
=> CANCELED [1/7] FROM docker.io/library/alpine@sha256:bc41182d7ef5ffc53a40b044e725193bc10142a1243f395ee852a8d  0.2s
=> => resolve docker.io/library/alpine@sha256:bc41182d7ef5ffc53a40b044e725193bc10142a1243f395ee852a8d9730fc2ad  0.0s
=> => sha256:bc41182d7ef5ffc53a40b044e725193bc10142a1243f395ee852a8d9730fc2ad 1.64kB / 1.64kB          0.0s
=> => sha256:1304f174557314a7ed9eddb4eab12fed12cb0cd9809e4c28f29af86979a3c870 528B / 528B          0.0s
=> => sha256:9c6f0724472873bb50a2ae67a9e7adcb57673a183cea8b06eb778dca859181b5 1.47kB / 1.47kB          0.0s
=> => sha256:213ec9aee27d8be045c6a92b7eac22c9a64b44558193775a1a7f626352392b49 0B / 2.81MB          0.0s
=> [internal] load build context                                                0.1s
=> => transferring context: 2B                                                  0.1s
=> CACHED [2/7] RUN apk update                                                  0.0s
=> CACHED [3/7] RUN apk add nodejs                                             0.0s
=> CACHED [4/7] RUN mkdir /app                                                 0.0s
=> ERROR [5/7] COPY index.js /app                                             0.0s
-----
> [5/7] COPY index.js /app:
-----
failed to compute cache key: "/index.js" not found: not found
```

```
anuvagarg@Anuvag-MacBook-Air Docker % docker build -f Dockerfile .
[+] Building 54.5s (6/11)
=> [internal] load build definition from Dockerfile                                0.1s
=> => transferring dockerfile: 37B                                              0.0s
=> [internal] load .dockerignore                                                 0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/alpine:latest                 2.1s
=> [1/7] FROM docker.io/library/alpine@sha256:bc41182d7ef5ffc53a40b044e725193bc10142a1243f395ee852a8d9730fc2ad  6.6s
=> => resolve docker.io/library/alpine@sha256:bc41182d7ef5ffc53a40b044e725193bc10142a1243f395ee852a8d9730fc2ad  0.0s
=> => sha256:bc41182d7ef5ffc53a40b044e725193bc10142a1243f395ee852a8d9730fc2ad 1.64kB / 1.64kB          0.0s
=> => sha256:1304f174557314a7ed9eddb4eab12fed12cb0cd9809e4c28f29af86979a3c870 528B / 528B          0.0s
=> => sha256:9c6f0724472873bb50a2ae67a9e7adcb57673a183cea8b06eb778dca859181b5 1.47kB / 1.47kB          0.0s
=> => sha256:213ec9aee27d8be045c6a92b7eac22c9a64b44558193775a1a7f626352392b49 2.81MB / 2.81MB          6.1s
=> => extracting sha256:213ec9aee27d8be045c6a92b7eac22c9a64b44558193775a1a7f626352392b49 0.4s
=> [internal] load build context                                                0.0s
=> => transferring context: 127B                                              0.0s
=> ERROR [2/7] RUN apk update                                             45.4s
-----
> [2/7] RUN apk update:
#5 0.444 fetch https://dl-cdn.alpinelinux.org/alpine/v3.16/main/x86_64/APKINDEX.tar.gz
#5 5.455 ERROR: https://dl-cdn.alpinelinux.org/alpine/v3.16/main: temporary error (try again later)
#5 5.457 fetch https://dl-cdn.alpinelinux.org/alpine/v3.16/community/x86_64/APKINDEX.tar.gz
#5 5.457 WARNING: Ignoring https://dl-cdn.alpinelinux.org/alpine/v3.16/main: No such file or directory
#5 45.34 v3.16.2-138-g7a2f8ae896 [https://dl-cdn.alpinelinux.org/alpine/v3.16/community]
#5 45.35 1 errors; 12113 distinct packages available
-----
executor failed running [/bin/sh -c apk update]: exit code: 1
```

2. Now we have dockerized the app, we will Build image from Dockerfile.

docker build -t hello:v0.1 .

```
anuvagarg@Anuvas-MacBook-Air Docker % docker build -t hello:v0.1 .
[+] Building 277.0s (12/12) FINISHED
=> [internal] load build definition from Dockerfile                                0.1s
=> => transferring dockerfile: 37B                                              0.0s
=> [internal] load .dockerignore                                                0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/alpine:latest                3.1s
=> [internal] load build context                                              0.0s
=> => transferring context: 29B                                                0.0s
=> CACHED [1/7] FROM docker.io/library/alpine@sha256:bc41182d7ef5ffc53a40b044e725193bc10142a1243f395ee852a8d97 0.0s
=> [2/7] RUN apk update                                                         33.1s
=> [3/7] RUN apk add nodejs                                                    239.1s
=> [4/7] RUN mkdir /app                                                         0.3s
=> [5/7] COPY index.js /app                                                    0.0s
=> [6/7] WORKDIR /app                                                         0.0s
=> [7/7] RUN node index.js                                                     0.3s
=> exporting to image                                                         0.4s
=> => exporting layers                                                         0.4s
=> => writing image sha256:96182b9da1e024809195b20fae468d290ecf629f48ada0c2b2594bac2642780f 0.0s
=> => naming to docker.io/library/hello:v0.1                                  0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

3. Create an account on Dockerhub and create a repository in it.

Public test repository: <https://hub.docker.com/repository/docker/anuvagarg/test-repo>

4. Tag the recently created image using the following command.

docker tag imageID Repositoryname

```
anuvagarg@Anuvas-MacBook-Air Docker % docker image ls
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
hello               v0.1        96182b9da1e0     5 minutes ago   52.7MB
redis               latest      3e42dd4e79c7     5 weeks ago     117MB
my-app              1.0         ca8cea303554     2 months ago    130MB
jenkins/jenkins     latest      9539c1956023     2 months ago    460MB
redis               <none>      dd8125f93b94     3 months ago    117MB
mongo               latest      1d3f6d5230f6     3 months ago    696MB
postgres            9.6         027ccf656dc1     6 months ago    200MB
docker/getting-started latest      bd9a9f733898     6 months ago    28.8MB
sonarqube            latest      4ac4842c584e     7 months ago    520MB
postgres            9.6.23     8f39a959063c     10 months ago   200MB
mongo-express        latest      2d2fb2cab8f      10 months ago   136MB
redis                4.0        191c4017dcdd     2 years ago     89.3MB

anuvagarg@Anuvas-MacBook-Air Docker % docker tag 96182b9da1e0 anuvagarg/test-repo
```

5. Login to Dockerhub from the console using the following command.

docker Login


```
anuvagarg@Anuvas-MacBook-Air Docker % docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: anuvagarg
Password:
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/
```


6. Now push the image on Docker Hub using the following command.


docker push Repositoryname

```
anuvagarg@Anuvas-MacBook-Air Docker % docker push anuvagarg/test-repo
Using default tag: latest
The push refers to repository [docker.io/anuvagarg/test-repo]
93fc773b6b74: Pushed
5f70bf18a086: Pushed
cbf5d993a075: Pushed
0c6fc74dfece: Pushed
5ed117662b75: Pushed
b0bd2703dca4: Pushed
994393dc58e7: Pushed
latest: digest: sha256:b605af68d6bc55c4992bbc946ddfa1e590640c8d4d5ddf70dd0a52dd4fd7f63a size: 1777
```

 **anuvagarg / test-repo**

Description


This repository does not have a description 

 Last pushed: 2 minutes ago



Docker commands [Public View](#)

To push a new tag to this repository,

```
docker push anuvagarg/test-repo:tagname
```

Tags and scans  **VULNERABILITY SCANNING - DISABLED** [Enable](#)

This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
 latest		---	2 minutes 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.

[Upgrade](#) [Learn more](#)

7. Pull and Run the container of your deployed image on docker hub.

Docker run -d image_name

```
anuvagarg@Anuvas-MacBook-Air Docker % docker run -d 96182b9da1e
f0f6f8696a8dc6ac8a551a9968beca92e22fa9e316468bed1081960e9aa0602e
```