Experiment no. 9

Aim: To learn Dockerfile instructions, build an image for a sample web application using Dockerfile.

Theory:

DOCKER

Docker is a set of platform as a service products that use OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software,
libraries and configuration files; they can communicate with each other through well-defined channels.

Images

An image is a read-only template with instructions for creating a Docker container. Often, an image is based *on* another image, with some additional customization. For example, you may build an image which is based on the ubuntu image, but installs the Apache web server and your application, as well as the configuration details needed to make your application run.

You might create your own images or you might only use those created by others and published in a registry. To build your own image, you create a Dockerfile with a simple syntax for defining the steps needed to create the image and run it. Each instruction in a Dockerfile creates a layer in the image. When you change the Dockerfile and rebuild the image, only those layers which have changed are rebuilt. This is part of what makes images so lightweight, small, and fast, when compared to other virtualization technologies.

FROM: This instruction is used to set the Base Image for the subsequent instructions. A valid Dockerfile must have FROM as its first instruction.
Ex. FROM ubuntu LABEL We can add labels to an image to organize images of our project. We need to use LABEL instruction to set label for the image. Ex. LABEL vendorl = "JavaTpoint".

RUN: RUN instruction allows you to install your application and packages requited for it. It executes any commands on top of the current image and creates a new layer by committing the results. Often you will find multiple RUN instructions in a Dockerfile.

CMD: CMD instruction allows you to set a default command, which will be executed only when you run container without specifying a command. If Docker container runs with a command, the default command will be ignored. If Dockerfile has more than one CMD instruction, all but last CMD instructions are ignored.

Rules: The source path must be inside the context of the build. We cannot COPY ../something /something because the first step of a docker build is to send the context directory (and subdirectories) to the docker daemon. If source is a directory, the entire contents of the directory are copied including filesystem metadata.

WORKDIR: The WORKDIR is used to set the working directory for any RUN, CMD and COPY instruction that follows it in the Dockerfile. If work directory does not exist, it will be created by default.

Output:

app.js pg.dockerfile

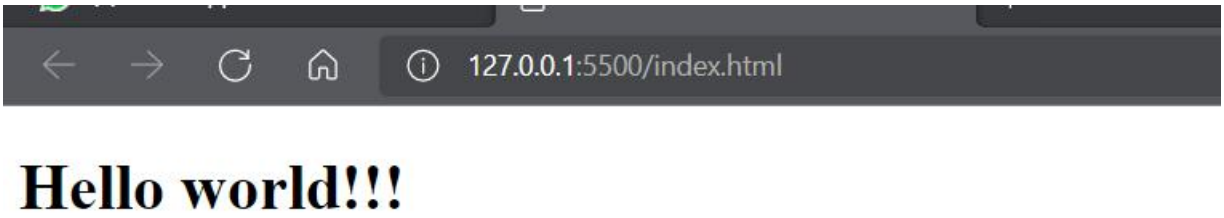
pg.dockerfile > ...

```
1  FROM node:alpine
2  COPY . /app
3  WORKDIR /app
4  CMD node app.js
```

```
$ docker run docker/whalesay cowsay Hello everyone
Unable to find image 'docker/whalesay:latest' locally
latest: Pulling from docker/whalesay
e190868d63f8: Pull complete
909cd34c6fd7: Pull complete
0b9bfabab7c1: Pull complete
a3ed95caeb02: Pull complete
00bf65475aba: Pull complete
c57b6bcc83e3: Pull complete
8978f6879e2f: Pull complete
8eed3712d2cf: Pull complete
Digest: sha256:178598e51a26abbc958b8a2e48825c90bc22e641de3d31e18aaf55f3258ba93b
Status: Downloaded newer image for docker/whalesay:latest
 _______________
< Hello everyone >
 ---------------
    \
     \
      \
                    ##        .
              ## ## ##       ==
           ## ## ## ##      ===
       /""""""""""""""""___/ ===
  ~~~ {~~ ~~~~ ~~~ ~~~~ ~~ ~ /  ===- ~~~
       \______ o          __/
        \     \        __/
          \____\______/
$
```

```
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
    PORTS               NAMES
$ ps -a
    PID TTY          TIME CMD
    155 pts/0    00:00:00 ps
$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
                    PORTS               NAMES
01537cce1567        docker/whalesay     "cowsay Hello everyo…"   About a minute ago   Exited (0)
 About a minute ago                     reverent_elion
$
```

Html:



Conclusion:
Hence we have successfully executed and implemented to build an image for a sample web application using Dockerfile.