Docker instruction

The RUN Docker instruction is used to execute commands during the image build process. It allows you to install packages, configure settings, and perform various actions within the container.

FROM: Specifies the base image for the Docker image. This is the starting point for building the container.

RUN: Executes commands in the container during the image build process. It is used to install software packages and set up the environment.

1.create the directory and change the directory and create the file:

```
ubuntu $ mkdir docker
ubuntu $ cd docker/
ubuntu $ touch dockerfile
```

Vi dockerfile

```
FROM nginx
RUN apt-get update ; apt-get install vim -y
COPY index.html /usr/share/nginx/html
```

2.Build the image:

```
ubuntu $ docker build -t nginx:v1 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future rele
ase.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  2.048kB
Step 1/2 : FROM nginx
latest: Pulling from library/nginx
f11c1adaa26e: Pull complete
c6b156574604: Pull complete
ea5d7144c337: Pull complete
1bbcb9df2c93: Pull complete
537a6cfe3404: Pull complete
767bff2cc03e: Pull complete
adc73cb74f25: Pull complete
Digest: sha256:67682bda769fae1ccf5183192b8daf37b64cae99c6c3302650f6f8bf5f0f95df
Status: Downloaded newer image for nginx:latest
```

```
 ---> fffffc90d343
Step 2/2 : RUN apt-get update ; apt-get install vim -y
 ---> Running in 76445fb94fd1
Get:1 http://deb.debian.org/debian bookworm InRelease [151 kB]
Get:2 http://deb.debian.org/debian bookworm-updates InRelease [55.4 kB]
Get:3 http://deb.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 Packages [8788 kB]
Get:5 http://deb.debian.org/debian bookworm-updates/main amd64 Packages [13.8 kB]
Get:6 http://deb.debian.org/debian-security bookworm-security/main amd64 Packages
 [168 kB]
Fetched 9224 kB in 1s (7611 kB/s)
Reading package lists...
Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  libgpm2 libsodium23 vim-common vim-runtime xxd
Suggested packages:
  gpm ctags vim-doc vim-scripts
The following NEW packages will be installed:
```

```
f) in auto mode
Processing triggers for libc-bin (2.36-9+deb12u7) ...
Removing intermediate container 76445fb94fd1
 ---> 3066b108acdd
Successfully built 3066b108acdd
Successfully tagged nginx:v1
```

3.Next will check the docker images:

```
ubuntu $ docker images
REPOSITORY   TAG       IMAGE ID       CREATED              SIZE
nginx        v1        3066b108acdd   About a minute ago   249MB
nginx        latest    fffffc90d343   2 weeks ago          188MB
```

4.Next will run the container with created images:

```
ubuntu $ docker run -d -p 8080:80 nginx:v1
4be6aea8b2c3e9dcac643df4fdd559ab8626de0046acb3a5d24de23f4fd714d1
```

5.will check the command docker ps -a

```
ubuntu $ docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED         STATUS
 PORTS                                  NAMES
4be6aea8b2c3   nginx:v1   "/docker-entrypoint.…"   6 seconds ago   Up 5 seconds
 0.0.0.0:8080->80/tcp, :::8080->80/tcp   inspiring_bassi
```

6.Next will inside the container and ID to bash it and again to check the to create:

```
ubuntu $ docker exec -it 4be6aea8b2c3 bash
root@4be6aea8b2c3:/# vi file45
```

COPY instruction copies new files or directories from and adds them to the filesystem of the container at the path

1.Create a directory and change directory create a docker file and Edit

```
ubuntu $ mkdir docker
ubuntu $ cd docker
ubuntu $ vi dockerfile
```

```
FROM nginx
RUN apt-get update ; apt-get install vim -y
COPY index.html /usr/share/nginx/html
```

Vi index.html file

```
ubuntu $ vi index.html
```

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}
form {border: 3px solid #f1f1f1;}

input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}
```

2.Next will build the image:

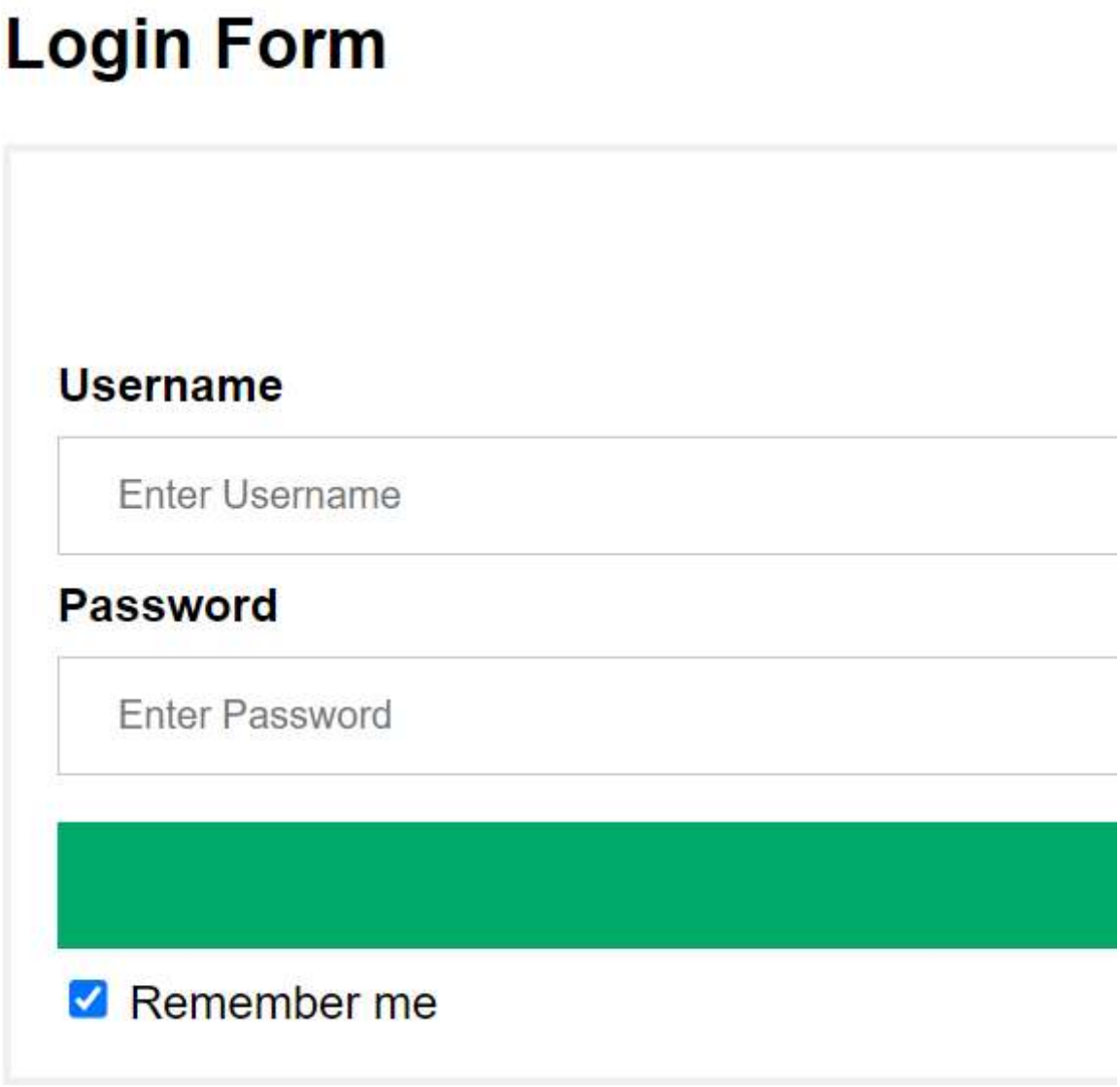
```
ubuntu $ docker build -t nginx:g2 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future rele
ase.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  4.608kB
Step 1/3 : FROM nginx
latest: Pulling from library/nginx
f11c1adaa26e: Pull complete
c6b156574604: Pull complete
ea5d7144c337: Pull complete
1bbcb9df2c93: Pull complete
537a6cfe3404: Pull complete
767bff2cc03e: Pull complete
adc73cb74f25: Pull complete
Digest: sha256:67682bda769fae1ccf5183192b8daf37b64cae99c6c3302650f6f8bf5f0f95df
Status: Downloaded newer image for nginx:latest
 ---> fffffc90d343
```

3.Created the container and run the image:

```
ubuntu $ docker run -d -p 8082:80 nginx:g2
8af54af84541bb0f31b8e5943297bb75501138b186101ebba4eb4fb4daf99f1f
```

4.Next Index.html file copied to container and view the homepage :



3. `LABEL` instruction adds metadata to an image.

   `MAINTAINER` instruction sets the Author field of the generated images.

1.Create a directory and change directory create a docker file and Edit

```
ubuntu $ mkdir docker
ubuntu $ cd docker
ubuntu $ vi dockerfile
```

Vi Dockerfile

```
FROM nginx
RUN apt-get update ; apt-get install vim -y
COPY index.html /usr/share/nginx/html
LABEL description='{{Gowthaman.Config.Labels}}'
LABEL maintainer="gowthaman"
```

2.Next will build the image:

```
ubuntu $ docker build -t nginx:g4 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future rele
ase.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  5.632kB
Step 1/5 : FROM nginx
 ---> fffffc90d343
Step 2/5 : RUN apt-get update ; apt-get install vim -y
 ---> Using cache
 ---> 3113358a22d8
Step 3/5 : COPY index.html /usr/share/nginx/html
 ---> Using cache
 ---> 213267a23634
Step 4/5 : LABEL description='{{Gowthaman.Config.Labels}}'
 ---> Using cache
 ---> 457dd3f43afc
Step 5/5 : LABEL maintainer="gowthaman"
 ---> Running in d3d7bb2e0703
```

```
 ---> Running in d3d7bb2e0703
Removing intermediate container d3d7bb2e0703
 ---> bbef3b7ab2e7
Successfully built bbef3b7ab2e7
Successfully tagged nginx:g4
```

3.Inspect and view the labels of docker image :

docker inspect nginx:v2

```
"OnBuild": null,
"Labels": {
    "description": "{{Gowthaman.Config.Labels}}",
    "maintainer": "gowthaman"
```

4.`ENV` instruction sets the environment variable:

`WORKDIR` instruction sets the working directory for any:

1.Create a directory and change directory create a docker file and Edit

```
ubuntu $ mkdir docker
ubuntu $ cd docker
ubuntu $ vi dockerfile
```

Vi Dockerfile

```
FROM nginx
RUN apt-get update ; apt-get install vim -y
LABEL description='{{Gowthaman.Config.Labels}}'
LABEL maintainer="gowthaman"
ENV MY_NAME="gowthaman"
WORKDIR /a
```

2.Next will build the image:

```
ubuntu $ docker build -t nginx:g3 .
```

3. Inspect and view the env of docker image :

```
            "Env": [
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bi
n",
                "NGINX_VERSION=1.27.0",
                "NJS_VERSION=0.8.4",
                "NJS_RELEASE=2~bookworm",
                "PKG_RELEASE=2~bookworm",
                "MY_NAME=gowthaman"
            ]
```

4.Inspect and view the workdir of docker image :

```
"Volumes": null,
"WorkingDir": "/a",
```

5.`CMD` instruction sets the command to be executed when running a container from an image.

`VOLUME` instruction creates a mount point:

1.Create a directory and change directory create a docker file and Edit

```
ubuntu $ mkdir docker
ubuntu $ cd docker
ubuntu $ vi dockerfile
```

Vi Dockerfile

```
FROM nginx
RUN apt-get update ; apt-get install vim -y
ENV MY_NAME="gowthaman"
WORKDIR /a
CMD ["sleep","infinity"]
VOLUME /vijay
```

2.Next will build the image:

```
ubuntu $ docker build -t nginx:g4  .
```

3.Inspect and view the CMD of docker image :

```
"Cmd": [
    "sleep",
    "infinity"
```

4.Inspect and view the volume of docker image :

```
"Volumes": {
    "/vijay": {}
},
```