DOCKERFILE ASSIGNMENT

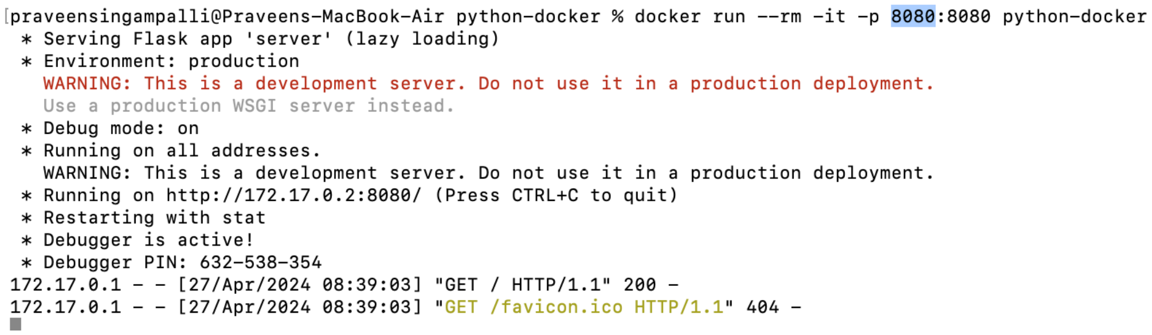
SCENARIO1 - PYTHON APP INTO DOCKER CONTAINER

Clone this code https://github.com/praveen1994dec/python-docker.git

Run the below dockerfile1

https://github.com/praveen1994dec/python-docker/blob/main/1-python-simple.Dockerfile

docker build -f 1-python-simple.Dockerfile -t python-docker .
docker run --rm -it -p 8080:8080 python-docker



```
[praveensingampalli@Praveens-MacBook-Air python-docker % docker run --rm -it -p 8080:8080 python-docker
 * Serving Flask app 'server' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://172.17.0.2:8080/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 632-538-354
172.17.0.1 - - [27/Apr/2024 08:39:03] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [27/Apr/2024 08:39:03] "GET /favicon.ico HTTP/1.1" 404 -
```

Run the below dockerfile2

https://github.com/praveen1994dec/python-docker/blob/main/2-python-hot-reload.Dockerfile

```
docker build -f 2-python-hot-reload.Dockerfile -t python-docker .
docker run --rm -it -p 8080:8080 python-docker
```

Run the below dockerfile3

https://github.com/praveen1994dec/python-docker/blob/main/4.python-ui-static.Dockerfile

```
docker build -f 4.python-ui-static.Dockerfile -t python-docker .
docker run --rm -it -p 8080:8080 python-docker
```

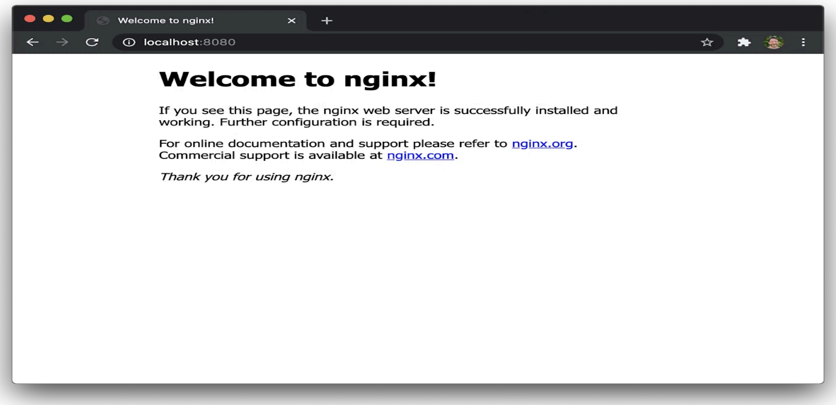
SCENARIO3- DOCKERFILE FOR GOLANG

```
# syntax=docker/dockerfile:1
FROM golang:1.21
WORKDIR /src
COPY <<EOF ./main.go
package main
import "fmt"
func main() {
 fmt.Println("hello, world")
}
EOF
RUN go build -o /bin/hello ./main.go
FROM scratch
COPY --from=0 /bin/hello /bin/hello
CMD ["/bin/hello"]
```

```
docker build -t golang_filename .
```

SCENARIO 4 - Running a basic web server using docker run command

```
docker run -it --rm -d -p 8080:80 --name web nginx
```



Welcome to nginx!

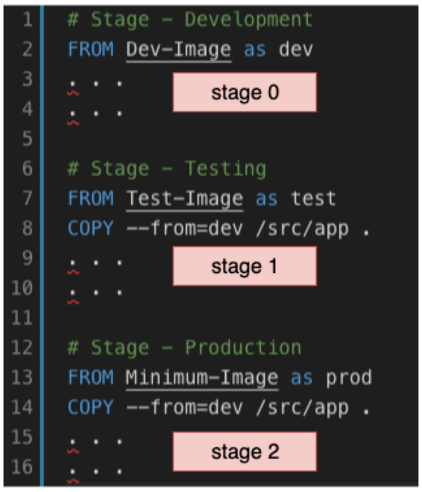
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

SCENARIO5-MULTI STAGE DOCKERFILE

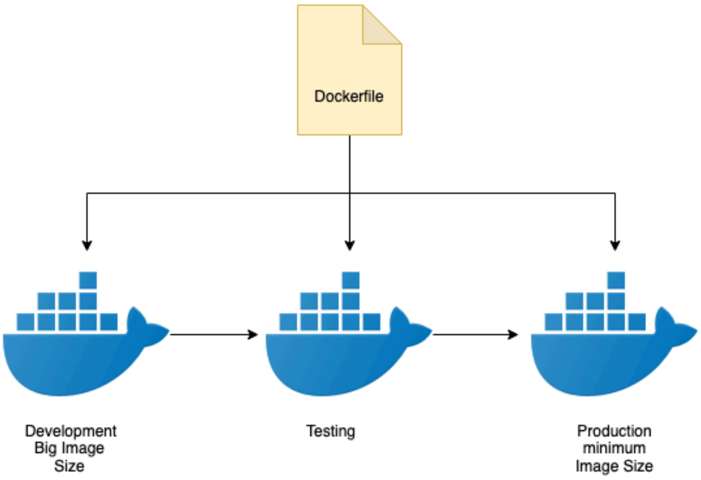
** Just for understanding



```
# Stage - Development
FROM Dev-Image as dev
. . .
. . .

# Stage - Testing
FROM Test-Image as test
COPY --from=dev /src/app .
. . .
. . .

# Stage - Production
FROM Minimum-Image as prod
COPY --from=dev /src/app .
. . .
. . .
```



```
# First stage: build the Java application <——---->
FROM maven:3.8.1-jdk-11-slim AS build
WORKDIR /app
COPY pom.xml .
RUN mvn dependency:go-offline
COPY src/ src/
RUN mvn package

# Second stage: copy the built app from the first stage <——---->
FROM openjdk:11-slim
```

```
WORKDIR /app
COPY --from=build /app/target/Maven-0.0.1-SNAPSHOT.jar .
EXPOSE 80
CMD ["java", "-jar", "Maven-0.0.1-SNAPSHOT.jar"]
```