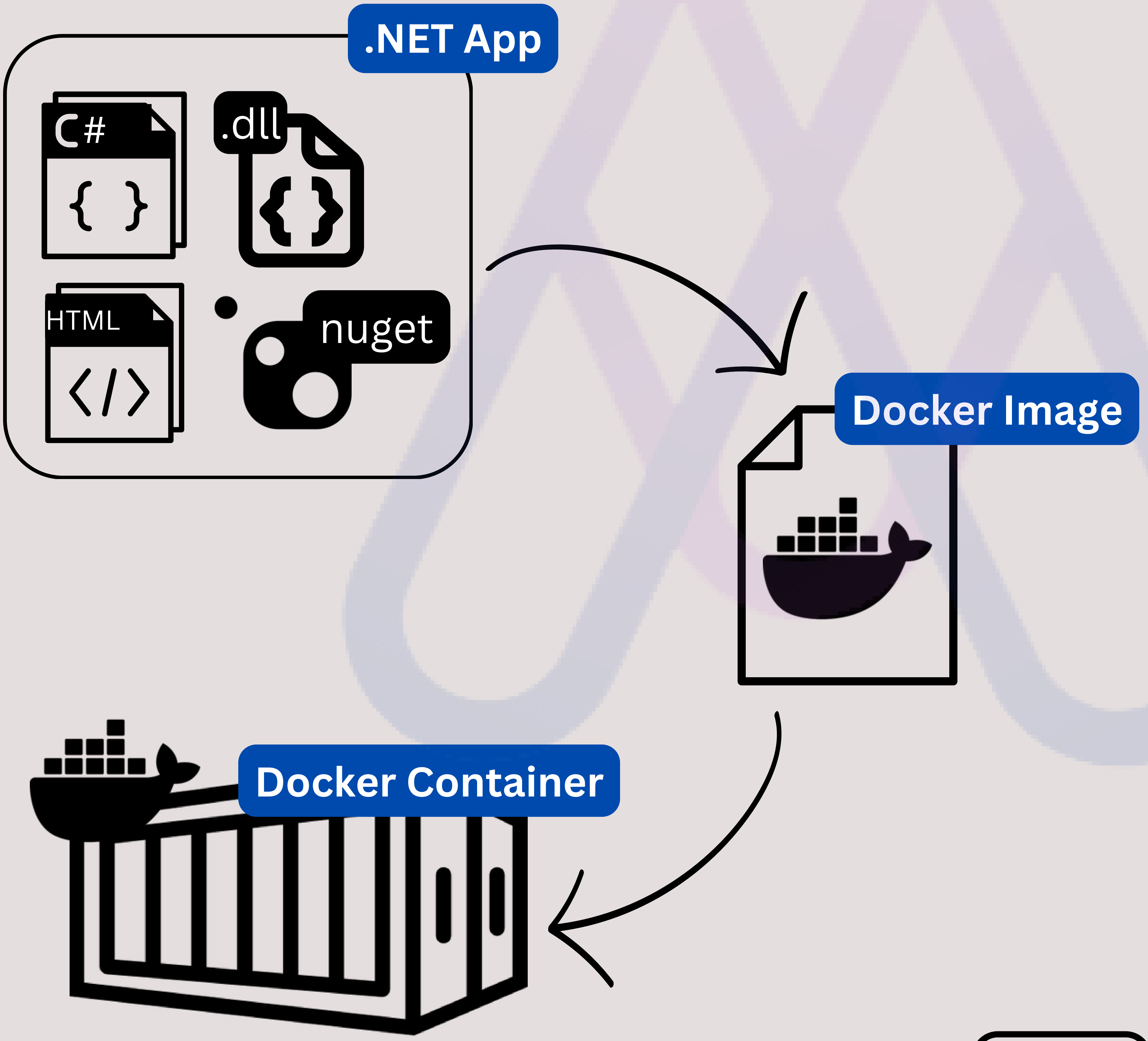
Ajay Patel

❤.NET

Time to get started

Still Not Using Docker?

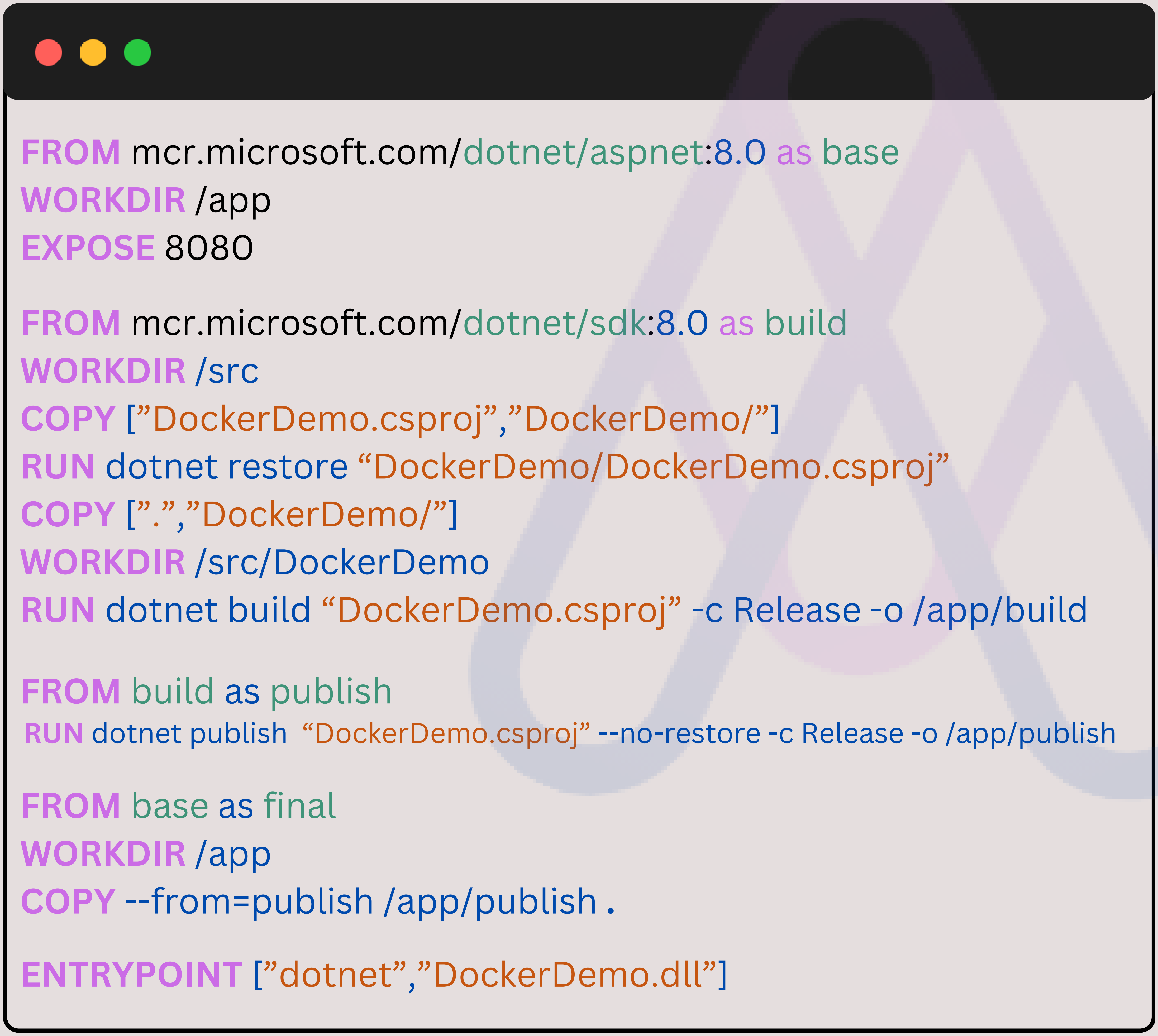


→

Ajay Patel

1. Create a Dockerfile

A Dockerfile includes instructions for building your Docker image.

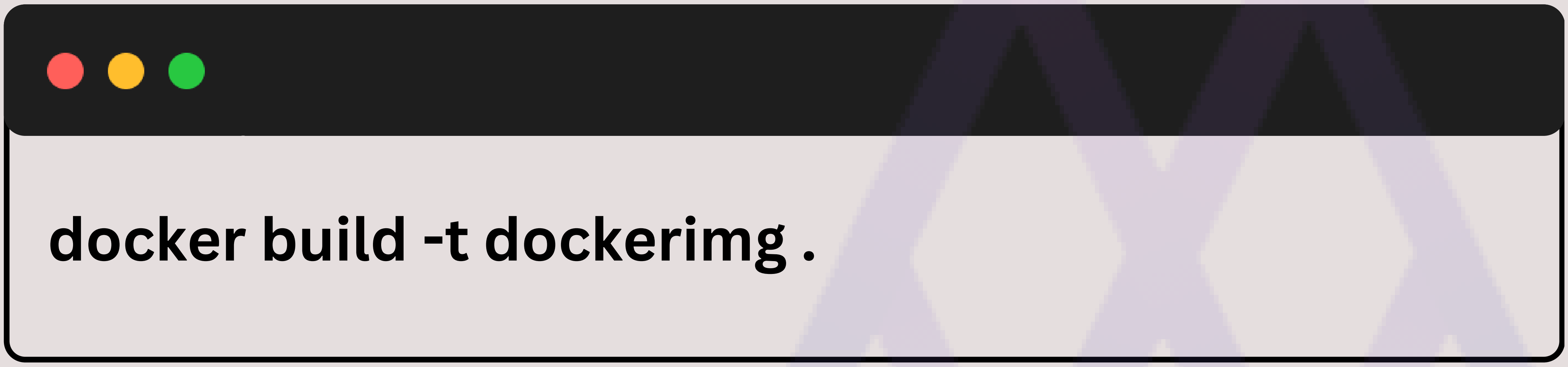


```
FROM mcr.microsoft.com/dotnet/aspnet:8.0 as base
WORKDIR /app
EXPOSE 8080

FROM mcr.microsoft.com/dotnet/sdk:8.0 as build
WORKDIR /src
COPY [”DockerDemo.csproj”,”DockerDemo/”]
RUN dotnet restore “DockerDemo/DockerDemo.csproj”
COPY [”.”,”DockerDemo/”]
WORKDIR /src/DockerDemo
RUN dotnet build “DockerDemo.csproj” -c Release -o /app/build

FROM build as publish
RUN dotnet publish  “DockerDemo.csproj” --no-restore -c Release -o /app/publish

FROM base as final
WORKDIR /app
COPY --from=publish /app/publish .

ENTRYPOINT [”dotnet”,”DockerDemo.dll”]
```

→

2. Build the Docker Image

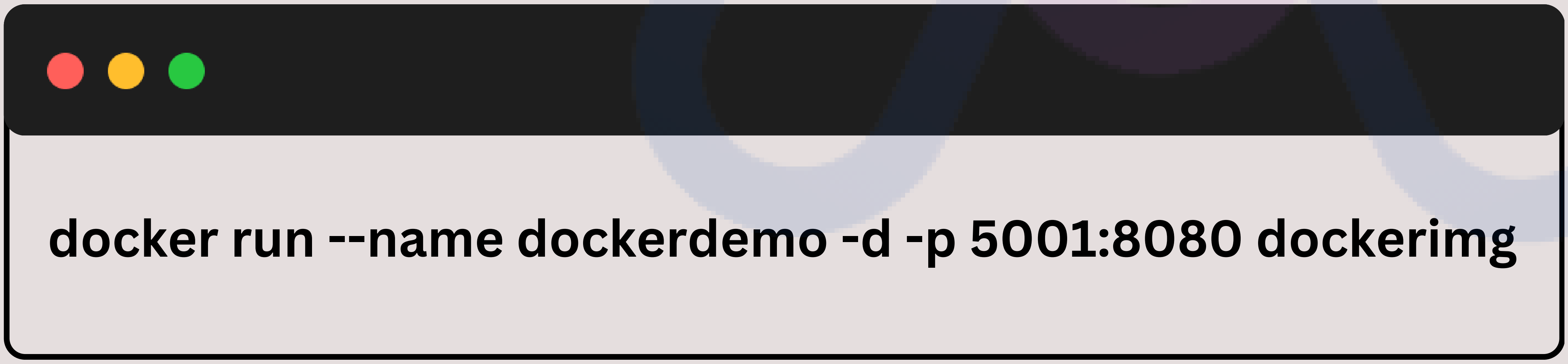
Execute this command in the terminal to create an image from your Dockerfile.



```
docker build -t dockerimg .
```

3. Run the Docker Container

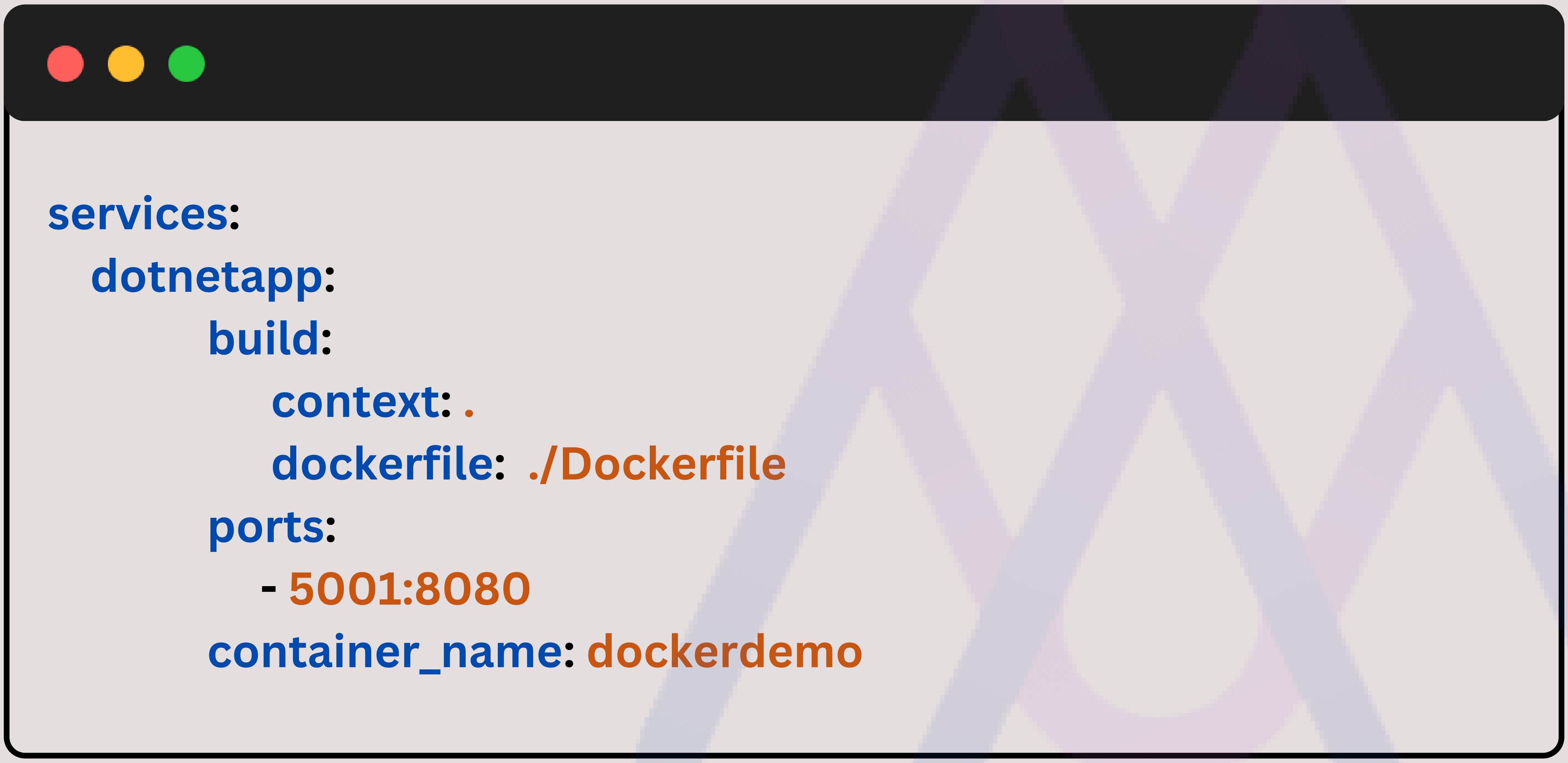
Execute this command to create and start the container.



```
docker run --name dockerdemo -d -p 5001:8080 dockerimg
```

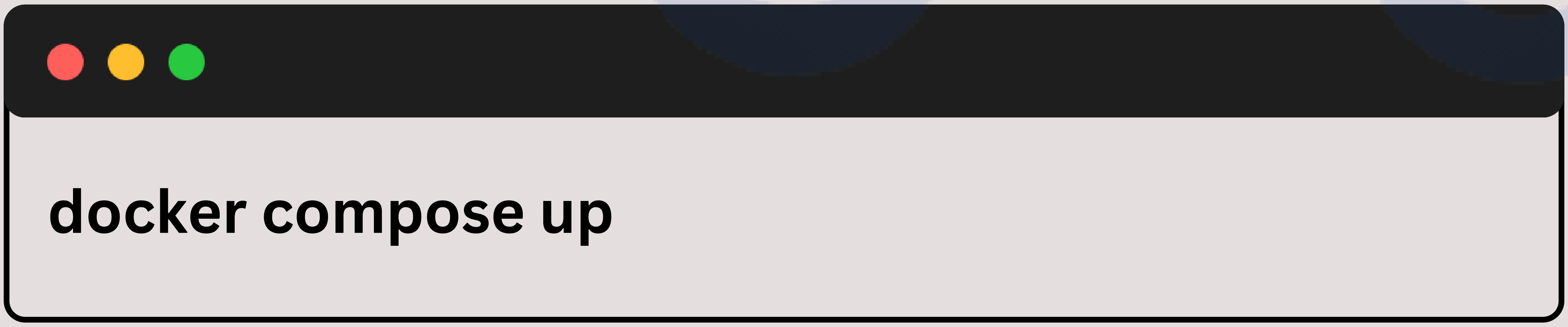
Docker Compose : Alternative to step 2 & 3

create docker-compose.yaml file allows you to build an image and start the container in single command. It is used to define and manage multi-container Docker applications.



```
services:
  dotnetapp:
        build:
            context: .
            dockerfile:  ./Dockerfile
        ports:
            - 5001:8080
        container_name: dockerdemo
```

Run below command to start your containerized application.



```
docker compose up
```

Ajay Patel

❤.NET

Knowledge is contagious, let’s spread it!

DO YOU LIKE THIS POST?

REPOST IT!

THANKS FOR READING