

Dockerfile Instructions

- Both ADD and COPY are used to copy files into image
 - ADD: it can copy from local to image as well as from urls
 - COPY: it can copy from local to image

Containerizing Nop Commerce

- [Refer Here](#) for the steps to create azure vm with docker installed
- NopCommerce requires dotnet 8
- dotnet core has two variants
 - sdk: developers
 - runtime: production
- dotnet core docker images [Refer Here](#)
- For the purpose of this lets use `mcr.microsoft.com/dotnet/sdk:8.0`
- The first version of the image to run nop commerce [Refer Here](#) for the changes
- [Refer Here](#) for changes to expose port 5000

```
learningthoughts@docker-machine:~/nop$ docker image build -t nop:1.1 .
[+] Building 0.7s (12/12) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 599B                             0.0s
=> [internal] load metadata for mcr.microsoft.com/dotnet/sdk:8.0 0.0s
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                    0.0s
=> [1/7] FROM mcr.microsoft.com/dotnet/sdk:8.0                 0.0s
=> [6/7] ADD https://github.com/nopSolutions/nopCommerce/releases/download/release-4.70.1/nopCommerce_4.70.1_NoSource_linux_x64.zip 0.1s
=> CACHED [2/7] RUN apt update                                  0.0s
=> CACHED [3/7] RUN apt install unzip -y                       0.0s
=> CACHED [4/7] RUN mkdir /app                                  0.0s
=> CACHED [5/7] WORKDIR /app                                    0.0s
=> CACHED [6/7] ADD https://github.com/nopSolutions/nopCommerce/releases/download/release-4.70.1/nopCommerce_4.70.1_NoSource_linux 0.0s
=> CACHED [7/7] RUN unzip nopCommerce_4.70.1_NoSource_linux_x64.zip && mkdir bin && mkdir logs 0.0s
=> exporting to image                                           0.1s
=> => exporting layers                                          0.0s
=> => writing image sha256:6689802898607706886f52a389f6fba2d023f7bb33a3b1578281663cde44765b 0.0s
=> => naming to docker.io/library/nop:1.1                     0.0s
learningthoughts@docker-machine:~/nop$
```

- Now build the image and run the container

```
learningthoughts@docker-machine:~/nop$ docker container run -d -P --name nop1 nop:1.1
33d4b2ba479670e23437a117339cb775c9c8dd52ffead0ec396a09ca5f0d133
learningthoughts@docker-machine:~/nop$ docker container ls
CONTAINER ID   IMAGE     COMMAND                  CREATED      STATUS      PORTS                               NAMES
33d4b2ba4796   nop:1.1   "dotnet Nop.Web.dll ..." 6 seconds ago Up 5 seconds 0.0.0.0:32768->5000/tcp, :::32768->5000/tcp  nop1
learningthoughts@docker-machine:~/nop$
```

English

nopCommerce installation

To complete this wizard, you must know some information regarding your database server ("connection string"). Please contact your ISP if necessary. If you're installing on a local machine or server, you might need information from your System Admin.

If you need information on how to use nopCommerce, visit [the documentation section on nopCommerce.com](#).

Store information

Admin user email:

Admin user password:

Confirm the password:

Country:

☐ Create sample data

☒ Subscribe to nopCommerce newsletter

- Now let's use aspnet runtime image as base image to run this application [Refer Here](#) for changes

```
learningthoughts@docker-machine:~/nop$ docker image build -t nop:2.0 .
[+] Building 20.0s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 268B
=> [internal] load metadata for mcr.microsoft.com/dotnet/aspnet:8.0-alpine-composite
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/4] FROM mcr.microsoft.com/dotnet/aspnet:8.0-alpine-composite@sha256:19d529bb70aa8002fa5d9a08dc29a58e2d3b96e672
=> [internal] load build context
=> => transferring context: 312.57MB
=> [2/4] RUN mkdir /app
=> [3/4] WORKDIR /app
=> [4/4] COPY app/ /app
=> exporting to image
=> => exporting layers
=> => writing image sha256:13f8c297188dc2db31ae096fb4050fe130c0f88e0252da3254c402f9f95b3ee1
=> => naming to docker.io/library/nop:2.0
learningthoughts@docker-machine:~/nop$ docker image ls
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
nop                  2.0         13f8c297188d  35 seconds ago 416MB
mcr.microsoft.com/dotnet/aspnet  8.0         672ee75ed306  2 days ago    217MB
mcr.microsoft.com/dotnet/aspnet  8.0-alpine3.19-composite  52571afbfff98  2 days ago    104MB
learningthoughts@docker-machine:~/nop$
```

- Now let's create a user and run the application as that user
- user:
 - name: nop
 - homedir: /app
 - shell: /bin/bash

- [Refer Here](#) for changeset

```
learningthoughts@docker-machine:~/nop$ docker image build -t nop:3.0 .
[+] Building 18.4s (9/9) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile              0.1s
=> => transferring dockerfile: 268B                             0.0s
=> [internal] load metadata for mcr.microsoft.com/dotnet/aspnet:8.0 0.0s
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                   0.0s
=> CACHED [1/4] FROM mcr.microsoft.com/dotnet/aspnet:8.0       0.0s
=> [internal] load build context                                0.6s
=> => transferring context: 301.91kB                             0.5s
=> [2/4] RUN useradd -d /app -m -s /bin/bash nop                1.1s
=> [3/4] WORKDIR /app                                          0.3s
=> [4/4] COPY app/ /app                                       12.2s
=> exporting to image                                          4.2s
=> => exporting layers                                          4.1s
=> => writing image sha256:8eab8265fb593177d25794331f725ff02189242f5d113b4ed43e4343c71aedd 0.0s
=> => naming to docker.io/library/nop:3.0                      0.0s
learningthoughts@docker-machine:~/nop$ docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nop                 3.0                8eab8265fb59       9 seconds ago      529MB
nop                 2.0                731bc1e088cc       16 minutes ago     529MB
mcr.microsoft.com/dotnet/aspnet 8.0                672ee75ed306       2 days ago         217MB
mcr.microsoft.com/dotnet/aspnet 8.0-alpine3.19-composite 52571afbfff98       2 days ago         104MB
learningthoughts@docker-machine:~/nop$
```

- run the container

Running Websites in a container

- Two popular mechanisms
 - Vanilla HTML, CSS, Javascript
 - SPA frameworks
 - angular
 - react
 - vue
- Lets take this template and run it inside container (we are containerizing web frontends) [Refer Here](#)
- Manual steps
 - download the zip file
 - unzip and copy the folder into html serving folder
- Docker experimentation steps

```
docker container run -d -P --name trail nginx:1.27
docker container cp /var/www/html/rent4u/ trail:/usr/share/nginx/html/rent4u
```

- Accessed the container port <http://<ip>:<forw-port>/rent4u>
- First version

```
FROM nginx:1.27
LABEL author="shaikkhajaibrahim"
ARG WEB_ZIP="https://www.free-css.com/assets/files/free-css-templates/download/page294/rent4u.zip"
RUN cd /tmp && curl -O ${WEB_ZIP}
RUN apt update \
  && apt install unzip -y \
  && cd /tmp && \
  unzip rent4u.zip && \
```

```
mv /tmp/rent4u-html /usr/share/nginx/html/rent4u
EXPOSE 80
# not writing CMD as i need command of base image
```

- Second version

```
FROM nginx:1.27
LABEL author="shaikkhajaibrahim"
COPY rent4u-html/ /usr/share/nginx/html/rent4u
EXPOSE 80
```

- Docker container modes
 - interactive
 - detached
 - foreground