Dockerfile

Best Practices

- Never run application inside docker container as root user.

Docker setup on AWS

- Creating AWS EC2 instance and installing docker Refer Here
- Prereq's:
  - Ensure aws cli is installed and configured
  - Ensure terraform
  - Ensure you have `id_rsa` and `id_rsa.pub` in your `~/.ssh`
- Execute the instructions as mentioned in the readme
- once your done practicing

```
terraform destroy
```

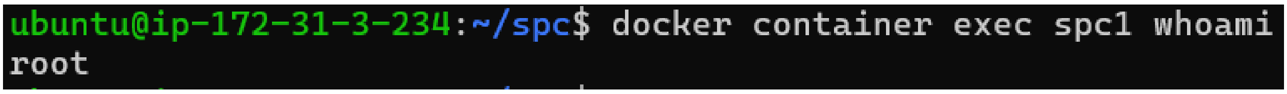
Docker setup on Azure

- Create azure vm with docker installed Refer Here
- Prereq's:
  - Ensure azure cli is installed and configured
  - Ensure terraform
  - Ensure you have `id_rsa` and `id_rsa.pub` in your `~/.ssh`
- Execute the instructions as mentioned in the readme
- once your done practicing

```
terraform destroy
```

Executing commands in running container

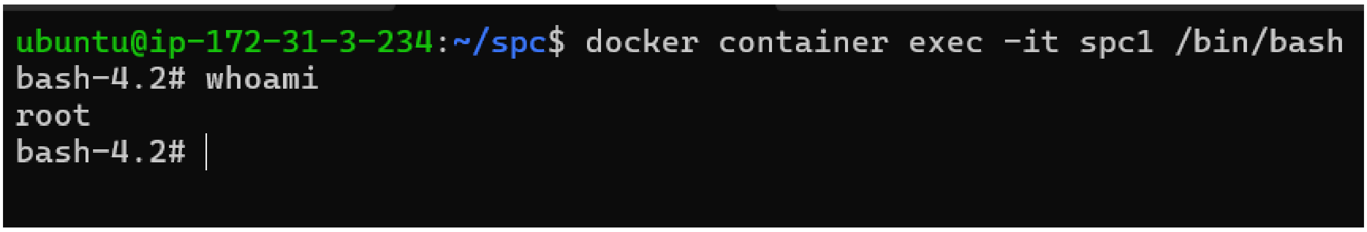
- docker container exec command helps in running commands in docker contianer



```
ubuntu@ip-172-31-3-234:~/spc$ docker container exec spc1 whoami
root
```

- if you want to get into container terminal

```
docker container exec -it <container-name> <terminal>
```

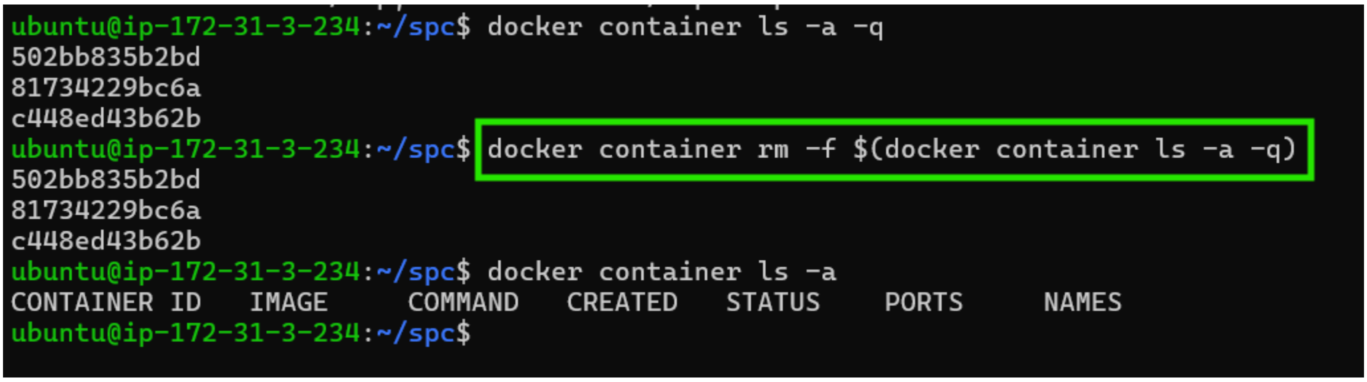


```
ubuntu@ip-172-31-3-234:~/spc$ docker container exec -it spc1 /bin/bash
bash-4.2# whoami
root
bash-4.2#
```

Removing contianers

- docker container rm [..]
- To remove all the containers

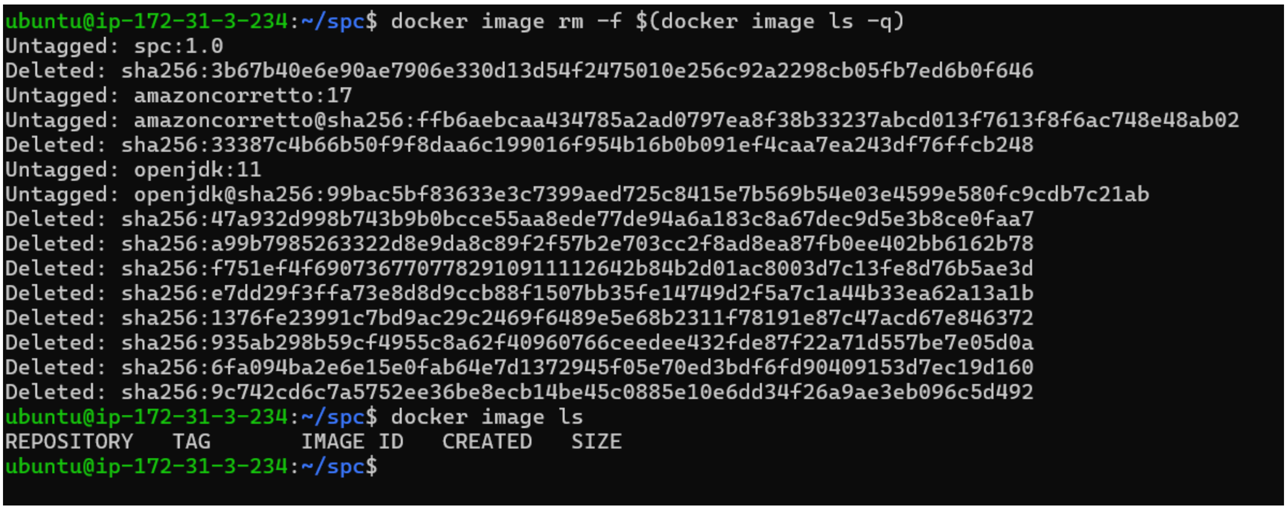
```
docker container rm -f $(docker container ls -a -q)
```



```
ubuntu@ip-172-31-3-234:~/spc$ docker container ls -a -q
502bb835b2bd
81734229bc6a
c448ed43b62b
ubuntu@ip-172-31-3-234:~/spc$ docker container rm -f $(docker container ls -a -q)
502bb835b2bd
81734229bc6a
c448ed43b62b
ubuntu@ip-172-31-3-234:~/spc$ docker container ls -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
ubuntu@ip-172-31-3-234:~/spc$
```

Removing image

- docker image rm [..]



```
ubuntu@ip-172-31-3-234:~/spc$ docker image rm -f $(docker image ls -q)
Untagged: spc:1.0
Deleted: sha256:3b67b40e6e90ae7906e330d13d54f2475010e256c92a2298cb05fb7ed6b0f646
Untagged: amazoncorretto:17
Untagged: amazoncorretto@sha256:ffb6aebcaa434785a2ad0797ea8f38b33237abcd013f7613f8f6ac748e48ab02
Deleted: sha256:33387c4b66b50f9f8daa6c199016f954b16b0b091ef4caa7ea243df76ffcb248
Untagged: openjdk:11
Untagged: openjdk@sha256:99bac5bf83633e3c7399aed725c8415e7b569b54e03e4599e580fc9cdb7c21ab
Deleted: sha256:47a932d998b743b9b0bcce55aa8ede77de94a6a183c8a67dec9d5e3b8ce0faa7
Deleted: sha256:a99b7985263322d8e9da8c89f2f57b2e703cc2f8ad8ea87fb0ee402bb6162b78
Deleted: sha256:f751ef4f6907367707782910911112642b84b2d01ac8003d7c13fe8d76b5ae3d
Deleted: sha256:e7dd29f3ffa73e8d8d9ccb88f1507bb35fe14749d2f5a7c1a44b33ea62a13a1b
Deleted: sha256:1376fe23991c7bd9ac29c2469f6489e5e68b2311f78191e87c47acd67e846372
Deleted: sha256:935ab298b59cf4955c8a62f40960766ceedee432fde87f22a71d557be7e05d0a
Deleted: sha256:6fa094ba2e6e15e0fab64e7d1372945f05e70ed3bdf6fd90409153d7ec19d160
Deleted: sha256:9c742cd6c7a5752ee36be8ecb14be45c0885e10e6dd34f26a9ae3eb096c5d492
ubuntu@ip-172-31-3-234:~/spc$ docker image ls
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
ubuntu@ip-172-31-3-234:~/spc$
```

Creating a new container in an interactive mode

- `docker container run -it <image:tag> <shell>`
-

Ensuring applications dont run as root user inside containers

- Consider the below Dockerfile

```
# take amazon correto 17 as a base image
FROM amazoncorretto:17
# add metadata
LABEL author="shaikkhajaibrahim"
LABEL project="lt-learning"
# download spring petclinic
RUN curl -O https://khajareferenceapps.s3.ap-south-1.amazonaws.com/spring-petclinic-3.2.0-SNAPSHOT.jar
# expose 8080 port as spring petclinic needs 8080
EXPOSE 8080
# command to start the application
CMD ["java", "-jar", "spring-petclinic-3.2.0-SNAPSHOT.jar"]
```

- Always add a user with necessary permissions to run your application.
- Adding user called as `spc` and giving permissions to run springpetclinic to spc
- The above is not possible with amazoncorretto as base image, so lets use exisiting user `nobody` to run spring petclinic
- USER instruction Refer Here
- WORKDIR instruction Refer Here
- Refer Here for the changes done to accomodate the user creation and switching.

Exercise: Run the springpetclinic application with openjdk:17 as base image

- Baseimage: `openjdk:17`
- user: `spc`
- working directory `/app`

Lets write a Dockerfile for nopCommerce

- Username: nop
- working directory: /app
- nopCommerce requires dotnet 8

Manually running nopCommerce on ubuntu linux

- install dotnet 8

```
sudo apt-get update && \
  sudo apt-get install -y dotnet-sdk-8.0 unzip
```

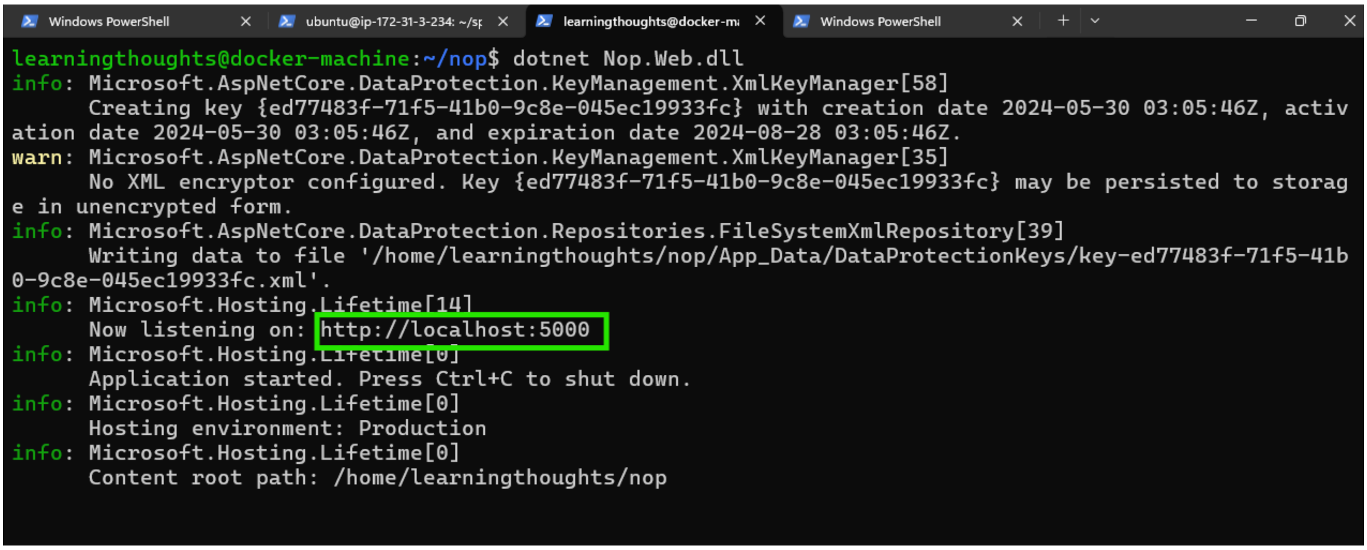
- download nopcommerce

```
mkdir nop
cd nop
wget https://github.com/nopSolutions/nopCommerce/releases/download/release-
```

```
4.70.1/nopCommerce_4.70.1_NoSource_linux_x64.zip
unzip nopCommerce_4.70.1_NoSource_linux_x64.zip
mkdir bin
mkdir logs
```

- execute the start command

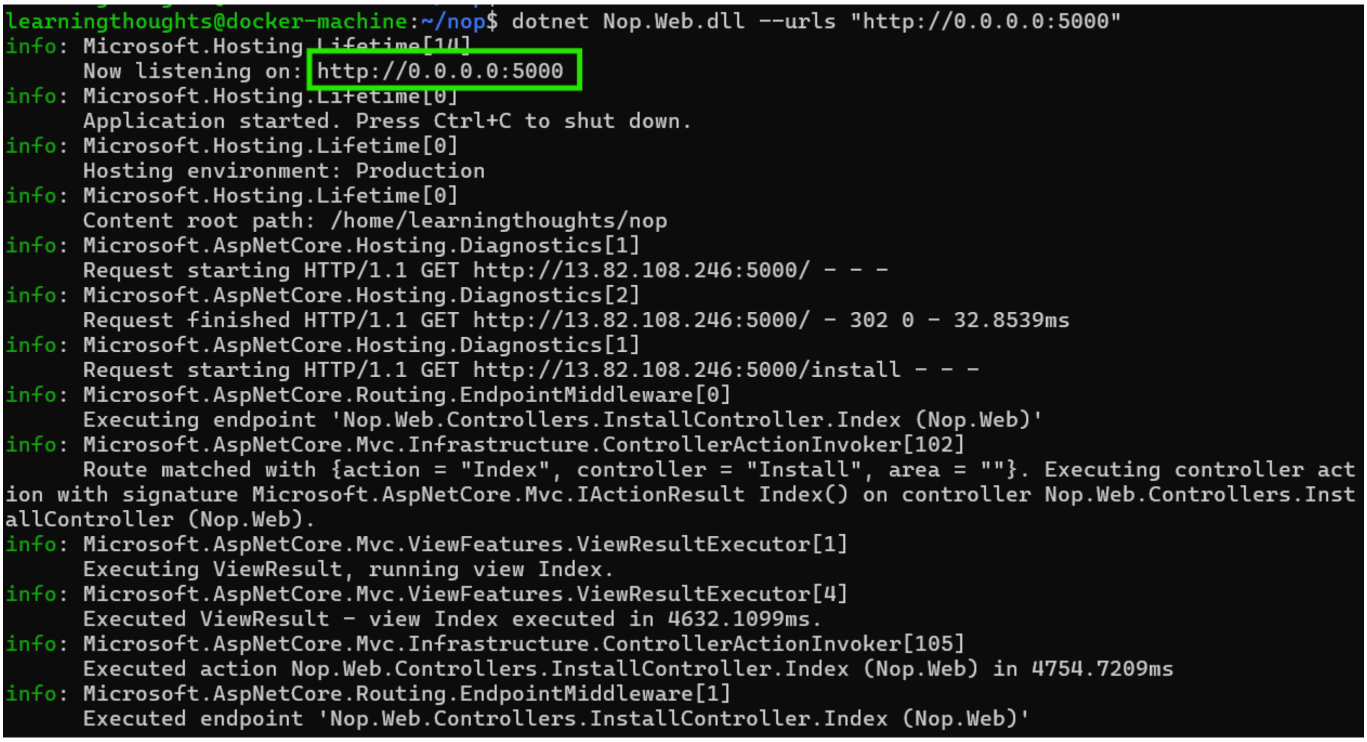
```
dotnet Nop.Web.dll
```



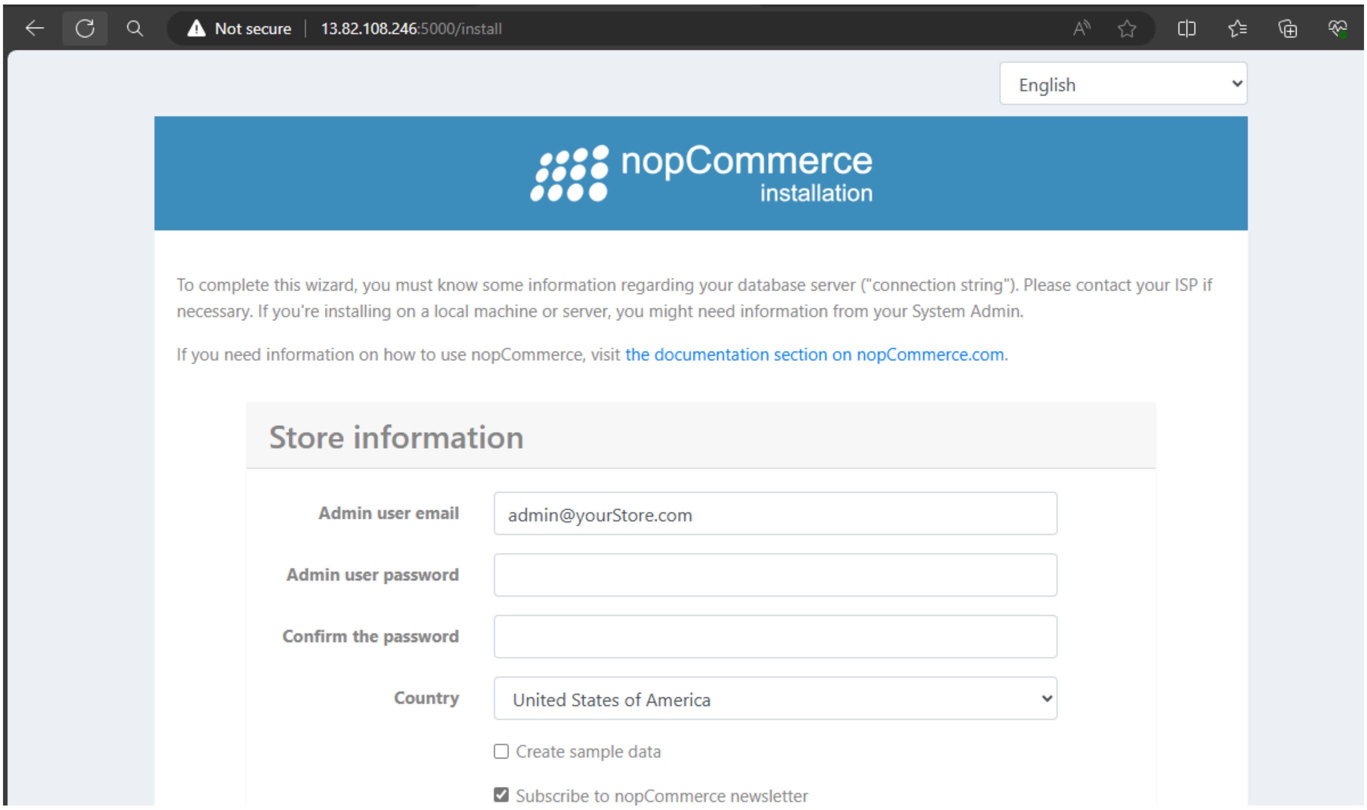
```
learningthoughts@docker-machine:~/nop$ dotnet Nop.Web.dll
info: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[58]
      Creating key {ed77483f-71f5-41b0-9c8e-045ec19933fc} with creation date 2024-05-30 03:05:46Z, activ
ation date 2024-05-30 03:05:46Z, and expiration date 2024-08-28 03:05:46Z.
warn: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[35]
      No XML encryptor configured. Key {ed77483f-71f5-41b0-9c8e-045ec19933fc} may be persisted to storag
e in unencrypted form.
info: Microsoft.AspNetCore.DataProtection.Repositories.FileSystemXmlRepository[39]
      Writing data to file '/home/learningthoughts/nop/App_Data/DataProtectionKeys/key-ed77483f-71f5-41b
0-9c8e-045ec19933fc.xml'.
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /home/learningthoughts/nop
```

- To make this application work on all interfaces

```
dotnet Nop.Web.dll --urls "http://0.0.0.0:5000"
```



```
learningthoughts@docker-machine:~/nop$ dotnet Nop.Web.dll --urls "http://0.0.0.0:5000"
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://0.0.0.0:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /home/learningthoughts/nop
info: Microsoft.AspNetCore.Hosting.Diagnostics[1]
      Request starting HTTP/1.1 GET http://13.82.108.246:5000/ - - -
info: Microsoft.AspNetCore.Hosting.Diagnostics[2]
      Request finished HTTP/1.1 GET http://13.82.108.246:5000/ - 302 0 - 32.8539ms
info: Microsoft.AspNetCore.Hosting.Diagnostics[1]
      Request starting HTTP/1.1 GET http://13.82.108.246:5000/install - - -
info: Microsoft.AspNetCore.Routing.EndpointMiddleware[0]
      Executing endpoint 'Nop.Web.Controllers.InstallController.Index (Nop.Web)'
info: Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker[102]
      Route matched with {action = "Index", controller = "Install", area = ""}. Executing controller act
ion with signature Microsoft.AspNetCore.Mvc.IActionResult Index() on controller Nop.Web.Controllers.Inst
allController (Nop.Web).
info: Microsoft.AspNetCore.Mvc.ViewFeatures.ViewResultExecutor[1]
      Executing ViewResult, running view Index.
info: Microsoft.AspNetCore.Mvc.ViewFeatures.ViewResultExecutor[4]
      Executed ViewResult - view Index executed in 4632.1099ms.
info: Microsoft.AspNetCore.Mvc.Infrastructure.ControllerActionInvoker[105]
      Executed action Nop.Web.Controllers.InstallController.Index (Nop.Web) in 4754.7209ms
info: Microsoft.AspNetCore.Routing.EndpointMiddleware[1]
      Executed endpoint 'Nop.Web.Controllers.InstallController.Index (Nop.Web)'
```



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@yourStore.com
Admin user password	
Confirm the password	
Country	United States of America

☐ Create sample data

☑ Subscribe to nopCommerce newsletter