**SSH to your AWS Workstation**

**ssh devops@<public-ip-addr**> of your Workstation  
Password is : **Dev0p$!!/**

**Replace <your-name> with your name throughout the lab.**

1. Clone the Git-repo.

|  |
| --- |
| $ mkdir /home/devops/dotnet $ cd /home/devops/dotnet/ $ git clone https://github.com/LovesCloud/dotnet-angular-docker.git $ cd dotnet-angular-docker/  $ dotnet clean $ dotnet build $ dotnet publish --self-contained -r linux-x64 -o dll |

2. Create a Dockerfile now.

|  |
| --- |
| $ vim Dockerfile |

Add the below lines in the docker file

|  |
| --- |
| **FROM microsoft/dotnet:2.2-runtime WORKDIR /data/ COPY dll/ /data/  RUN chmod +x angular\* EXPOSE 5001 EXPOSE 5000 EXPOSE 80 ENTRYPOINT ["./angular"]** |

**Save and exit the vim editor.**

3. Run the below commands to build a Docker Container from the git repo.

|  |
| --- |
| $ docker build . -t <your-name>-docker.net |

4. Check the Image after it has been build.

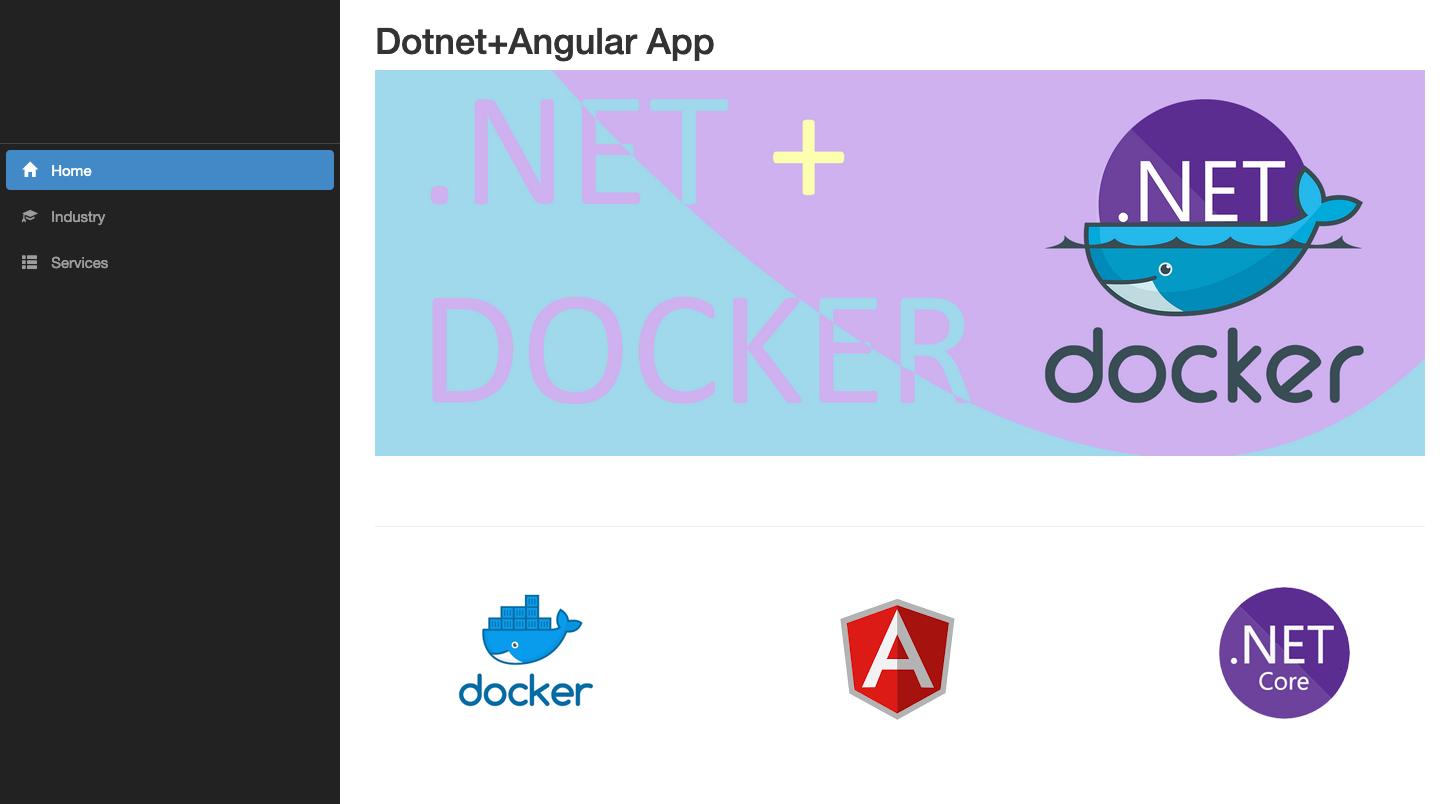
|  |
| --- |
| $ docker images |

5. Run the docker container from the build image

|  |
| --- |
| $ docker run -d -p 80:80 --name <your-name>-dotnet-app <your-name>-docker.net  $ docker ps |

The dotnet Docker Container running on port 80, and can be accessed from the public IP of the AWS Workstation on default Port 80

**http://<ip-addr-aws-workstation>**\



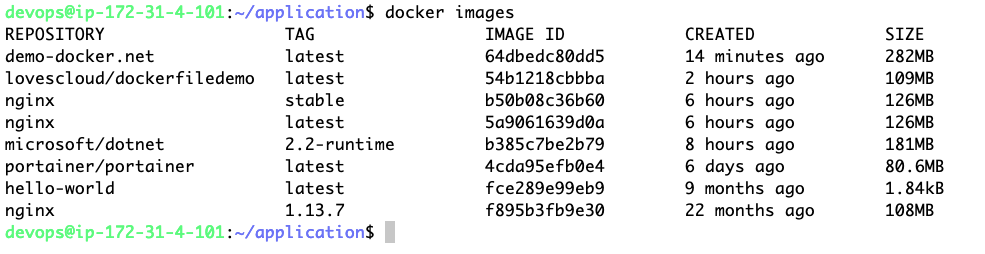
**STOP THE CONTAINER BEFORE PROCEEDING TO THE NEXT LAB**

6. Stop the docker container by running the below command.

|  |
| --- |
| $ docker stop <your-name>-dotnet-app |

7. Tag and Push the Dotnet Docker Image

|  |
| --- |
| $ docker images |



8. Tag the Docker image and push to docker-hub.

|  |
| --- |
| $ docker tag <Image-ID> yourhubusername/dockerfiledemo:latest |

Example

