**Setting Up Jenkins Pipeline to Deploy Docker Swarm Source Code**

**ProductBackendApplication.java:**

package com;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication(scanBasePackages = "com")

public class ProductBackendApplication {

public static void main(String[] args) {

SpringApplication.run(ProductBackendApplication.class, args);

}

}

Product.java

**package** com.bean;

**public** **class** Product {

**private** **int** pid;

**private** String pname;

**private** **float** price;

**private** String url;

**public** **int** getPid() {

**return** pid;

}

**public** **void** setPid(**int** pid) {

**this**.pid = pid;

}

**public** String getPname() {

**return** pname;

}

**public** **void** setPname(String pname) {

**this**.pname = pname;

}

**public** **float** getPrice() {

**return** price;

}

**public** **void** setPrice(**float** price) {

**this**.price = price;

}

**public** String getUrl() {

**return** url;

}

**public** **void** setUrl(String url) {

**this**.url = url;

}

@Override

**public** String toString() {

**return** "Product [pid=" + pid + ", pname=" + pname + ", price=" + price + ", url=" + url + "]";

}

**public** Product(**int** pid, String pname, **float** price, String url) {

**super**();

**this**.pid = pid;

**this**.pname = pname;

**this**.price = price;

**this**.url = url;

}

**public** Product() {

**super**();

// **TODO** Auto-generated constructor stub

}

}

ProductController.java

**package** com.controller;

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.springframework.http.MediaType;

**import** org.springframework.web.bind.annotation.CrossOrigin;

**import** org.springframework.web.bind.annotation.GetMapping;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RestController;

**import** com.bean.Product;

@RestController

@RequestMapping("product")

@CrossOrigin()

**public** **class** ProductController {

@GetMapping(value = "allProduct",produces = MediaType.***APPLICATION\_JSON\_VALUE***)

**public** List<Product> getAllProduct() {

List<Product> listOfProduct = **new** ArrayList<Product>();

listOfProduct.add(**new** Product(1, "TV", 850000, "https://cdn1.smartprix.com/rx-iSUL96nlp-w1200-h1200/SUL96nlp.jpg"));

listOfProduct.add(**new** Product(2, "Computer", 350000, "https://image.shutterstock.com/image-photo/modern-desktop-computer-wireless-keyboard-260nw-55573504.jpg"));

listOfProduct.add(**new** Product(3, "Laptop", 970000, "https://encrypted-tbn2.gstatic.com/shopping?q=tbn:ANd9GcRrgpCYffiSTw3gl3XFtgC\_ZLXN5eG0J-WFWDXk58Eyobs-NY6eY0p7gEh9URgogg\_MxGc0VZjQ6Ltfi6aSayw5clWdGwlLhVlK\_2NZfS\_9WMMbzPBZ2\_7b4LxqVAj0v8uHfXXFN2-g&usqp=CAc"));

listOfProduct.add(**new** Product(4, "PenDrive",1200, "https://cdn.pixabay.com/photo/2015/07/20/19/50/usb-853230\_\_340.png"));

**return** listOfProduct;

}

}

Dockerfile

FROM openjdk:8

COPY target/product-backend.jar .

CMD ["java","-jar","product-backend.jar"]

Pom.xml

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.6.12</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.example</groupId>

<artifactId>Product-backend</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>Product-backend</name>

<description>Demo project for Spring Boot</description>

<properties>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<finalName>product-backend</finalName>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

Product.component.html:

<div>

    <table border="1">

        <tr>

            <th>PId</th>

            <th>PName</th>

            <th>Price</th>

            <th>Url</th>

        </tr>

        <tr \*ngFor="let product of products">

            <td>{{product.pid}}</td>

            <td>{{product.pname}}</td>

            <td>{{product.price}}</td>

            <td><img src={{product.url}} width="200" height="200"/></td>

        </tr>

    </table>

</div>

Product.component.ts:

import { Component, OnInit } from '@angular/core';

import { Product } from '../product';

import { ProductService } from '../product.service';

@Component({

  selector: 'app-product',

  templateUrl: './product.component.html',

  styleUrls: ['./product.component.css']

})

export class ProductComponent implements OnInit {

  products:Array<Product>=[];

  constructor(public ps:ProductService) { }

  ngOnInit(): void {

    this.loadAllProduct();

  }

  loadAllProduct(){

    //this.ps.loadAllProduct().subscribe(data=>this.Products=data,error=>console.log(error),()=>console.log("completed"));

    this.ps.loadAllProduct().subscribe({

      next:(result:any)=>this.products=result,

      error:(error:any)=>console.log(error),

      complete:()=>console.log("completed")

      });

  }

}

app.component.html:

<div>

  <app-product></app-product>

</div>

app.module.ts:

import { NgModule } from '@angular/core';

import { BrowserModule } from '@angular/platform-browser';

import {HttpClientModule} from '@angular/common/http'

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { ProductComponent } from './product/product.component';

@NgModule({

  declarations: [

    AppComponent,

    ProductComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,HttpClientModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

product.service.ts:

import { HttpClient } from '@angular/common/http';

import { Injectable } from '@angular/core';

import { Observable } from 'rxjs';

import { Product } from './product';

@Injectable({

  providedIn: 'root'

})

export class ProductService {

  constructor(public http:HttpClient) {  }

  loadAllProduct():Observable<Product[]>{

    return this.http.get<Product[]>("http://localhost:9091/product/allProduct");

  }

}

product.ts:

export class Product {

    constructor(public pid:number,

        public pname:string,

        public price:number,

        public url:string){}

}

Dockerfile:

FROM nginx

COPY dist/product-frontend/ /usr/share/nginx/html

Docker-compose.yml:

version: "3"

services:

  product-backend:

    image: product-app

    ports:

      - "9091:9091"

    networks:

      - product-management-system

  product-frontend:

    image: product-frontend-app

    ports:

      - "80:80"

    depends\_on:

      - product-backend

    networks:

      - product-management-system

networks:

  product-management-system:

DockerSwarm:

Docker-compose.yml:

 version: "3"

services:

  product-backend:

    image: omsreeja/product-app:1.0

    ports:

      - "9091:9091"

    networks:

      - product-management-system

  product-frontend:

    image: omsreeja/product-frontend-app:1.0

    ports:

      - "80:80"

    depends\_on:

      - product-backend

    networks:

      - product-management-system

networks:

  product-management-system: