|  |
| --- |
| **[문항1] 1. Docker 이미지 생성하고, 컨테이너 실행, dockerhub 등록하시오**  1-1. sample springboot project작성(30)  Controller 작성  package com.MavenProject.demo.controller;  import org.springframework.http.HttpStatus; import org.springframework.http.ResponseEntity; import org.springframework.web.bind.annotation.GetMapping; import org.springframework.web.bind.annotation.RestController;  @RestController public class HelloController { //http://localhost:8080/hello   @GetMapping("/hello") public ResponseEntity<Object> hello() { String result = "Hello, Docker!";  return new ResponseEntity<>(result, HttpStatus.*OK*); } }  -----------------------------------------------------------------------------------  1-2. docker image : springboot-sample1(30)  Dockerfile 작성  #https://hub.docker.com/\_/openjdk #open jdk 17 FROM openjdk:17-alpine ARG *JAR\_FILE*=build/libs/\*.jar #build된 파일중jar 생성(libs)  #파일 복사 COPY ${*JAR\_FILE*} agaroup.jar #환경 설정 ENTRYPOINT ["java", "-jar", "-Dspring.profiles.active=prod", "/agaroup.jar"]  -----------------------------------------------------------------------------------  1-3. docker conatiner: springboot-app(30)  - 프로젝트 생성  docker run --name springboot-sample5 --net=springboot-sample5 -dit -p 8088:80 -e WORDPRESS\_DB\_PRESS\_DB\_HOST=mysql\_ex3 wordpress:latest  - 이미지 빌드  docker build -t springboot-sample5 .    - 컨테이너 생성 및 실행  docker run --name springboot-app -p 8080:8080 papepopo/spring-sample5    ----------------------------------------------------------------------------------- 1-4. dockerhub등록 및 run(10) |

- Docker Hub 로그인

docker login

- Docker Hub 이미지 태그 지정

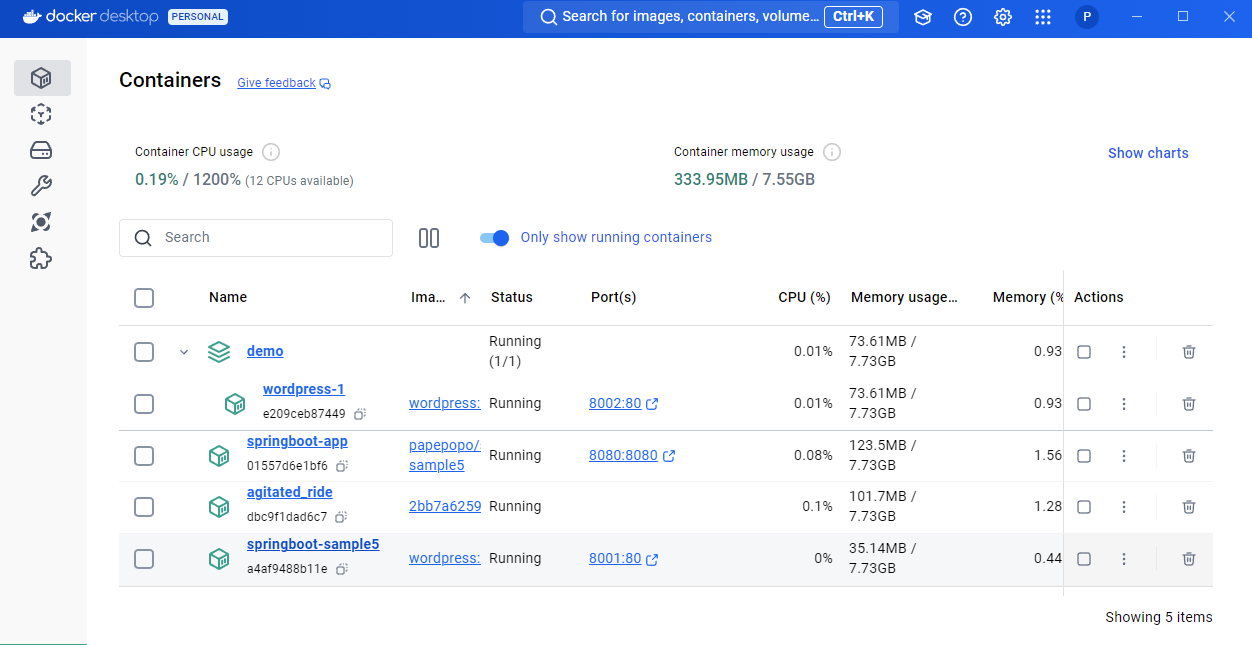
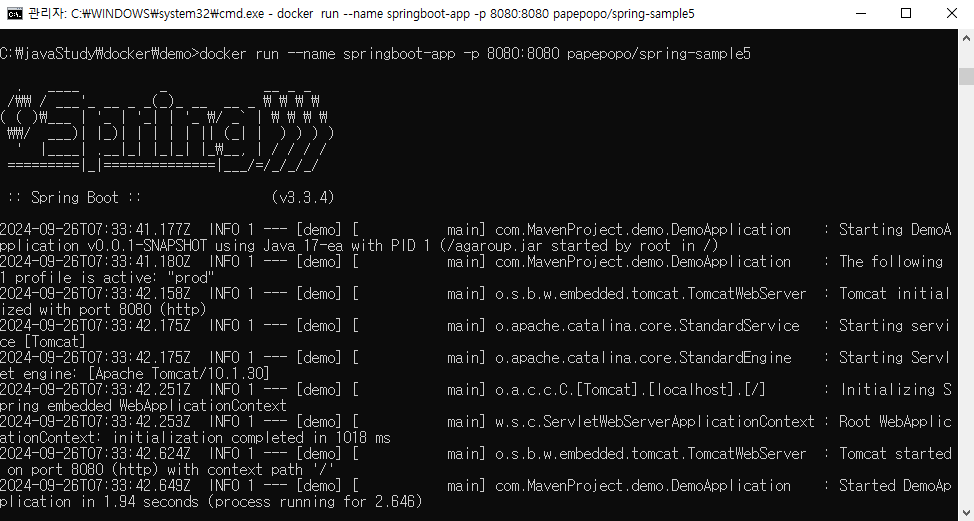
docker tag springboot-sample1 your\_username/springboot-sample1

- Docker Hub 이미지 push

docker push your\_username/springboot-sample1

- Docker Hub 이미지 실행

docker run --name springboot-app -p 8080:8080 papepopo/spring-sample5



- Dockerhub 이미지 업로드 페이지

<https://hub.docker.com/u/papepopo>

