**Spring – Docker – linux 연동**

**명령어 정리**

**linux**

pwd : 현재경로

cd : 경로이동

ls : 현재경로에 존재하는 파일/폴더 확인

파일 실행 : ./파일명

권한 설정 : chmod 777 파일명

폴더 모든 파일 권한 설정 : chmod -r 777 폴더명

**vim 명령어**

파일열기 : vi 파일명

입력모드로 전환 : i

명령모드로 전환 : esc

저장 : w

나가기 : q

저장하고 나가기 : wq

저장하지 않고 나가기 : q!

**docker 명령어**

**pull** : docker pull <저장소명/이미지명:태그>

**push** : docker push <(push 할)저장소명/이미지명:태그> <(dockhub의)저장소명/이미지명:태그>

**commit** : docker commit <컨테이너이름 or 컨테이너id > <저장할 이미지명>

**실행** : \*이미지로부터 컨테이너를 생성

docker run -d --name <컨테이너 이름> -p <로컬포트:도커포트> <이미지이름 or 이미지id>

-d : 백그라운드로 실행

**도커 실행 후 쉘로 돌아오기** : ctrl c

**컨테이너 시작** : 중지된 컨테이너를 시작 docker start <컨테이너이름 or 컨테이너id >  
**컨테이너 재부팅** : docker restart <컨테이너이름 or 컨테이너id>  
**컨테이너 중지** : docker stop <컨테이너이름 or 컨테이너id >  
**실행중인 컨테이너로 접근 하기** : docker exec -it <container\_id> bash / **돌아오기** : control z  
**실행 중인 컨테이너 조회** : docker ps / **중지된 컨테이너도 모두 조회** : docker ps –a  
**컨테이너 중지** : docker stop <컨테이너이름 or 컨테이너id >  
**컨테이너 삭제** : docker rm <컨테이너이름 or 컨테이너id >

**이미지 확인** : docker images  
**이미지 삭제** : docker rmi <이미지명 or 이미지id >  
**이미지 이름 변경** :   
docker image tag <기존 이미지명>:<기존 태그명> <새 이미지명>:<새 태그명>

**진행**

0. maven 설치

* 1. <https://maven.apache.org/download.cgi> 다운로드
  2. 원하는 위치에 압축 해제하기
  3. 시스템변수 path에 압축 해제한 위치 추가
  4. 프로젝트 pom.xml파일 maven-compiler-plugin의 configuration에

 <encoding>UTF-8</encoding> 추가

F. 프로젝트 내 src/test/java 아래 패키지 삭제

1. docker 설치

2. docker-desktop 설치

3. tomcat 이미지 pull하기

4. tomcat 이미지 실행하기 + 운영체제 버전확인 cat /etc/issue

5. tomcat:latest 컨테이너에 vim 설치하기

**APT** : 리눅스는 오픈소스이기 때문에 운영체제 버전이 다양하다.

APT는 debian 기반 리눅스에서 사용하는 패키지관리도구이다. APT를 사용해 원하는 패키지(프로그램)을 다운 받을 수 있다.

**vim** : 텍스트편집기

#apt-get update

#apt-get upgrade

#apt-get install vim

\* tomcat 실행 시 ojdbc6을 못 찾아서 발생하는 에러 해결

톰캣이 스스로 jar파일을 scan해서 발생하는 문제

tomcat의 context.xml 파일에

<JarScanner scanClassPath="false" /> 추가해 scan 옵션을 종료

6. 컨테이너 수정 내용을 base-tomcat:1.0 이미지로 commit

여기까지 했으면 docker 쪽 준비는 끝.

**프로젝트**

7. maven으로 docker 이미지를 빌드 할 때 태그에 대문자가 있으면 docker에서 에러가 발생한다. (docker가 대문자를 허용하지 않음)

아래 태그의 모든 대문자를 소문자로 변경해준다.

<modelVersion>4.0.0</modelVersion>

<groupId> com.kh</groupId>

<artifactId> spring\_docker</artifactId>

<name> spring\_docker</name>

<packaging>war</packaging>

<version>1.0.0</version>

8. pom.xml에 docker maven plugin 코드 작성  
<https://github.com/spotify/docker-maven-plugin>

<!-- Docker maven plugin -->

<plugin>

<groupId>com.spotify</groupId>

<artifactId>docker-maven-plugin</artifactId>

<configuration>

<!—생성할 이미지이름-->

<!— docker 계정명/이미지이름 -->

<imageName>azimemory/spring-docker</imageName>

<!—기반 이미지로 사용할 이미지. 톰켓 실행환경 위에 배포할 예정-->

<baseImage>base-tomcat:1.0</baseImage>

<resources>

<resource>

<!-- 배포할 경로-->

<targetPath>/usr/local/tomcat/webapps/</targetPath>

<!-- 배포할 리소스 경로-->

<!-- >${project.basedir} 메이븐 환경변수 프로젝트 경로-->

<!-- <http://maven.apache.org/guides/introduction/introduction-to-the-pom.html> 참고-->

<directory>${project.basedir}/target/</directory>

<!-- war파일명 project.build.finalName으로 지정-->

<!-- artifactId + version 으로 지어진다.-->

<include>${project.build.finalName}.war</include>

</resource>

</resources>

</configuration>

</plugin>

9. mavne으로 docker이미지 생성

mvn clean package docker:build

10. 이미지 실행, 컨테이너 콘솔로 접근해 server.xml에서 contextPath 변경

<Host> 태그 안에 아래의 태그 추가

<Context path= "/" docBase="war파일명" relodable="false"></Context>

11. 컨테이너 commit 이미지명 docker계정/spring-docker:1.0

12. docker hub 로그인 한 뒤

docker push docker계정/spring-docker:1.0

13. docker hub 확인

**Oracle VM Instance**

1. putty 설치
2. puttygen으로 Type of key to generate:RSA로 지정 후 ssh key 생성
3. VM 인스턴스 생성

**SSH키 추가**에 2번에서 생성한 public key 파일등록

1. Putty를 통해서 vm 인스턴스 접속
2. VM 인스턴스 root 계정 password 변경 하기 : sudo passwd root
3. Root 계정으로 로그인 : su -
4. VM 인스턴스에 docker 설치

yum -y update

yum -y install docker docker-registry

부팅 시 자동 실행을 하도록 서비스 등록

systemctl enable docker.service

도커 실행

systemctl start docker.service

1. 도커 로그인 : docker login
2. Docker로 image pull받기
3. 이미지 실행
4. 도메인 생성하기 <https://www.freenom.com/en/index.html?lang=en>

DNS방식으로 생성할 것!