**Kakao I Cloud VM 1Tier WAS-DB 시스템 구축**

**201735871 장건**

**- 목적**

1Tier 기반의 VM 환경에서 WAS와 DB를 연동하여 연동 결과를 {공인IP:WAS포트/~.jsp} 형태의 경로로 확인. (jsp 파일이 실행되면서 Connection 문구 출력)

**- 범위**

이번 레포트에서는 3Tier 기반으로 연동하는 것이 아닌 1Tier기반, 연동에 필요한 포트만 열고 진행.

**- 설정**

------------------------- VM 환경 -------------------------

OS : CentOS 7.9(64bit)

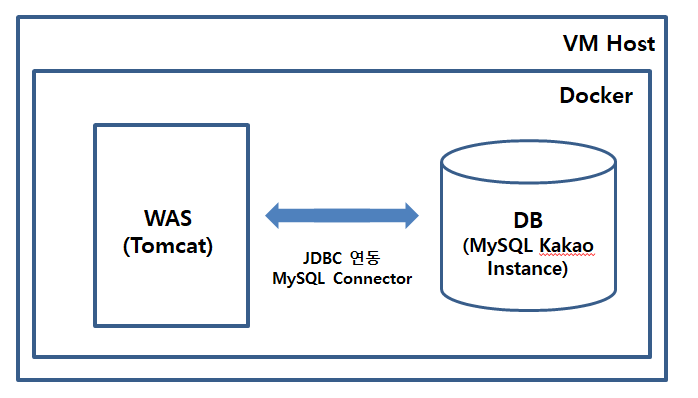
인스턴스 타입 : A1-2-STD (vCPU : 2개, Memory : 8GB)

------------------------- 시스템 구축 조건 -------------------------

WAS : Tomcat

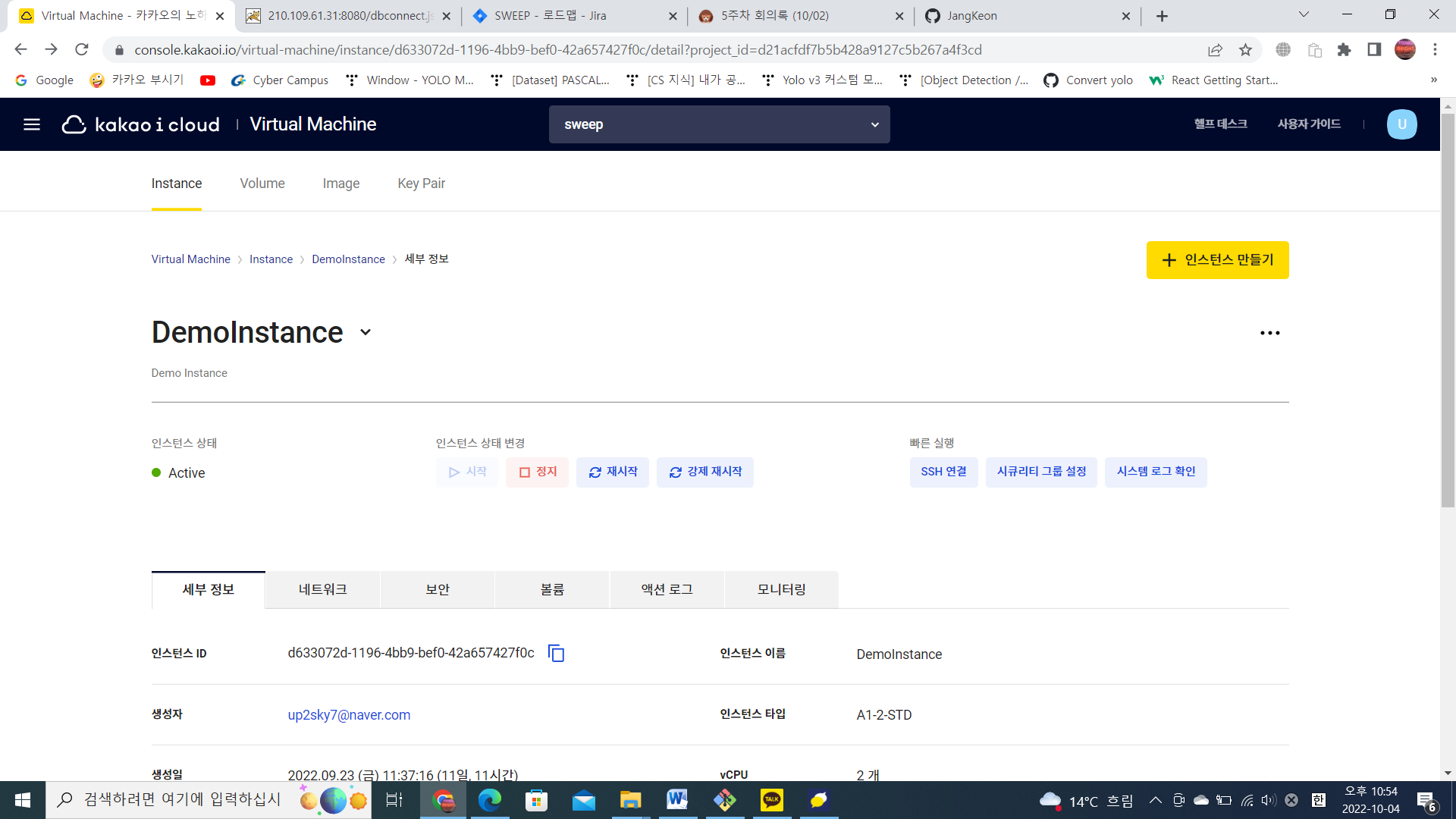
DB : MySQL (Kakao Instance)

**- 시스템 아키텍처 구성도**



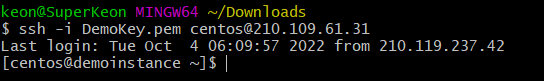
**- 구축 과정**

**1. Kakao I Cloud에서 VM 생성 (**[**[KiC Hands-on] 1. VM 기반의 웹 서비스 구성하기 (oopy.io)**](https://kic-gachon.oopy.io/f156508a-b53c-4bcf-be2c-4072fac8efdf) **– Step1 따라하기.)**



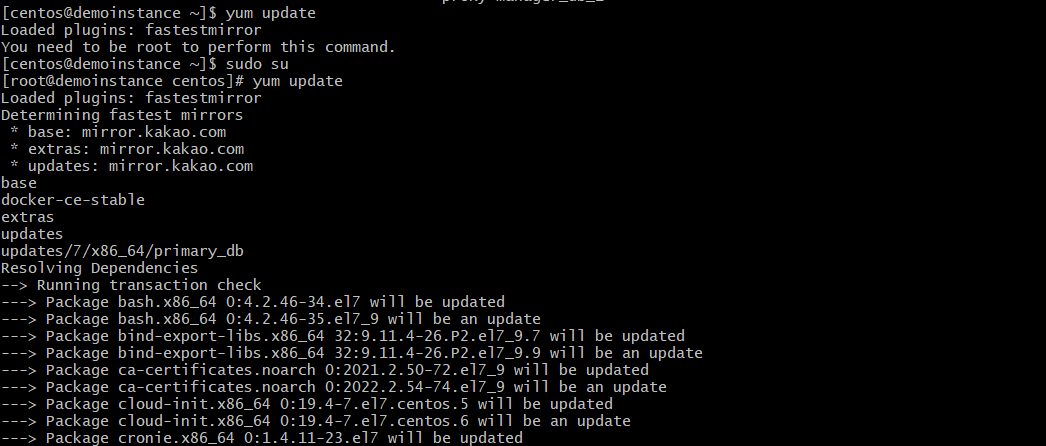
**2. Git bash 를 통해 생성한 VM에 SSH 접근**

ssh -i ${KEYPAIR\_NAME}.pem centos@${VM\_PUBLIC\_IP}



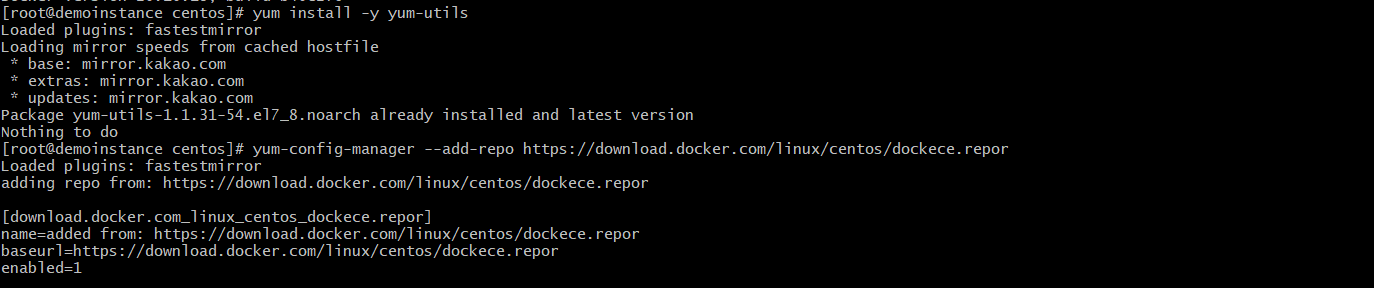
**3. 리눅스 패키지 업데이트, docker 설치 확인, docker 레포지토리 시스템에 추가, docker 실행**

yum –y update

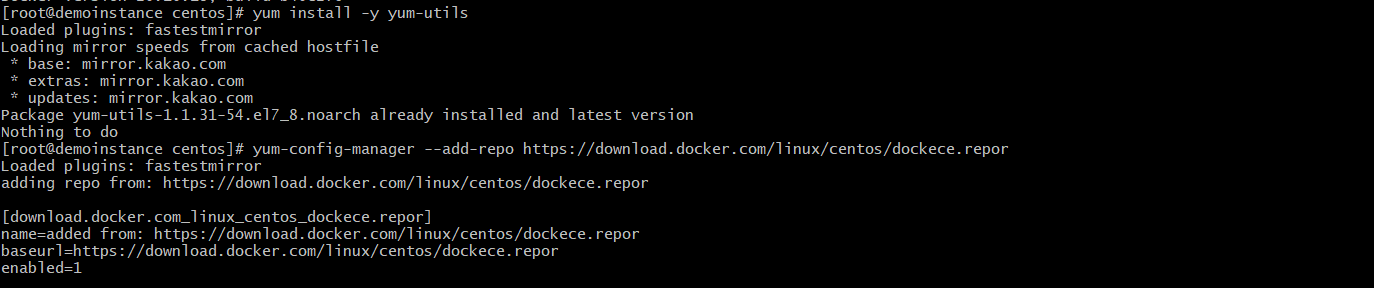


docker –version

yum install –y yum-utils



yum-config-manager –add-repo https://download.docker.com/linux/centos/docker-ce.repo



Systemctl start docker

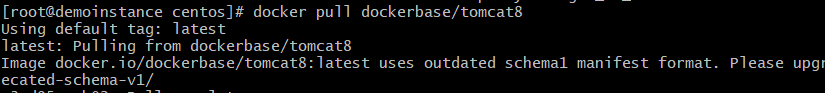
Systemctl enable docker

Systemctl status docker

**4. WAS 구축**

Web Application Server로 Tomcat을 설치

Docker pull dockerbase/tomcat8

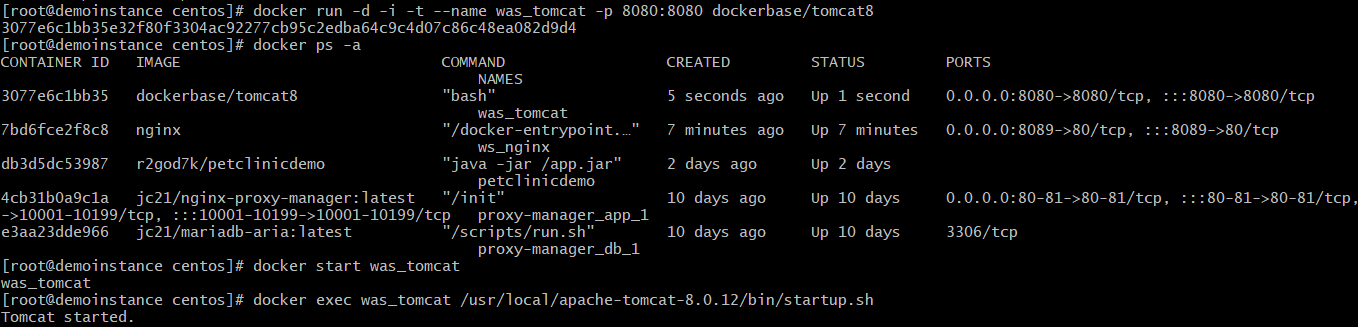


Docker run –d –I –t –name was\_tomcat –p 8080:8080 dockerbase/tomcat8

Docker ps –a

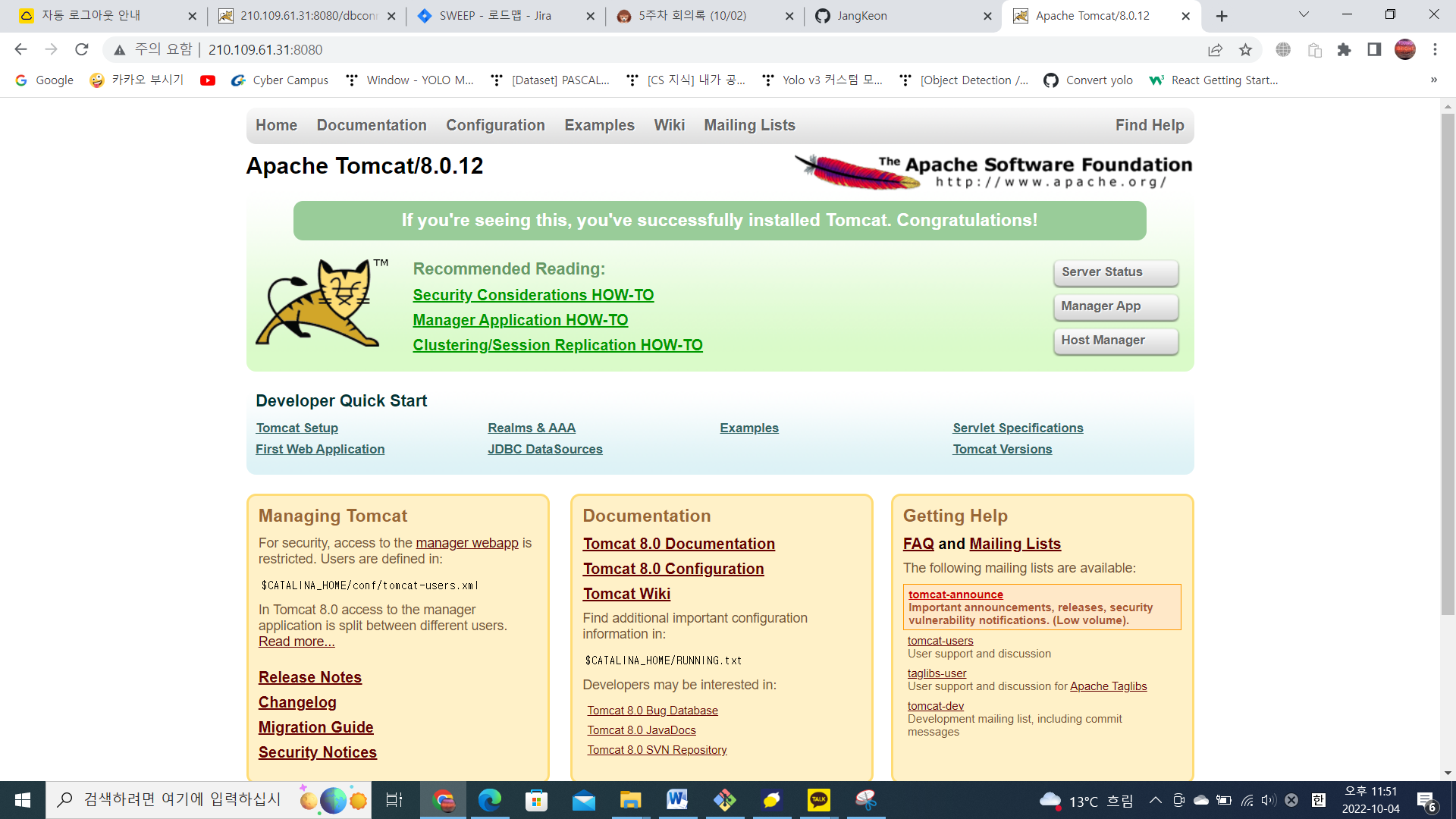
Docker start was\_tomcat

Docker exec was\_tomcat /usr/local/apache-tomcat-8.0.12/bin/startup.sh



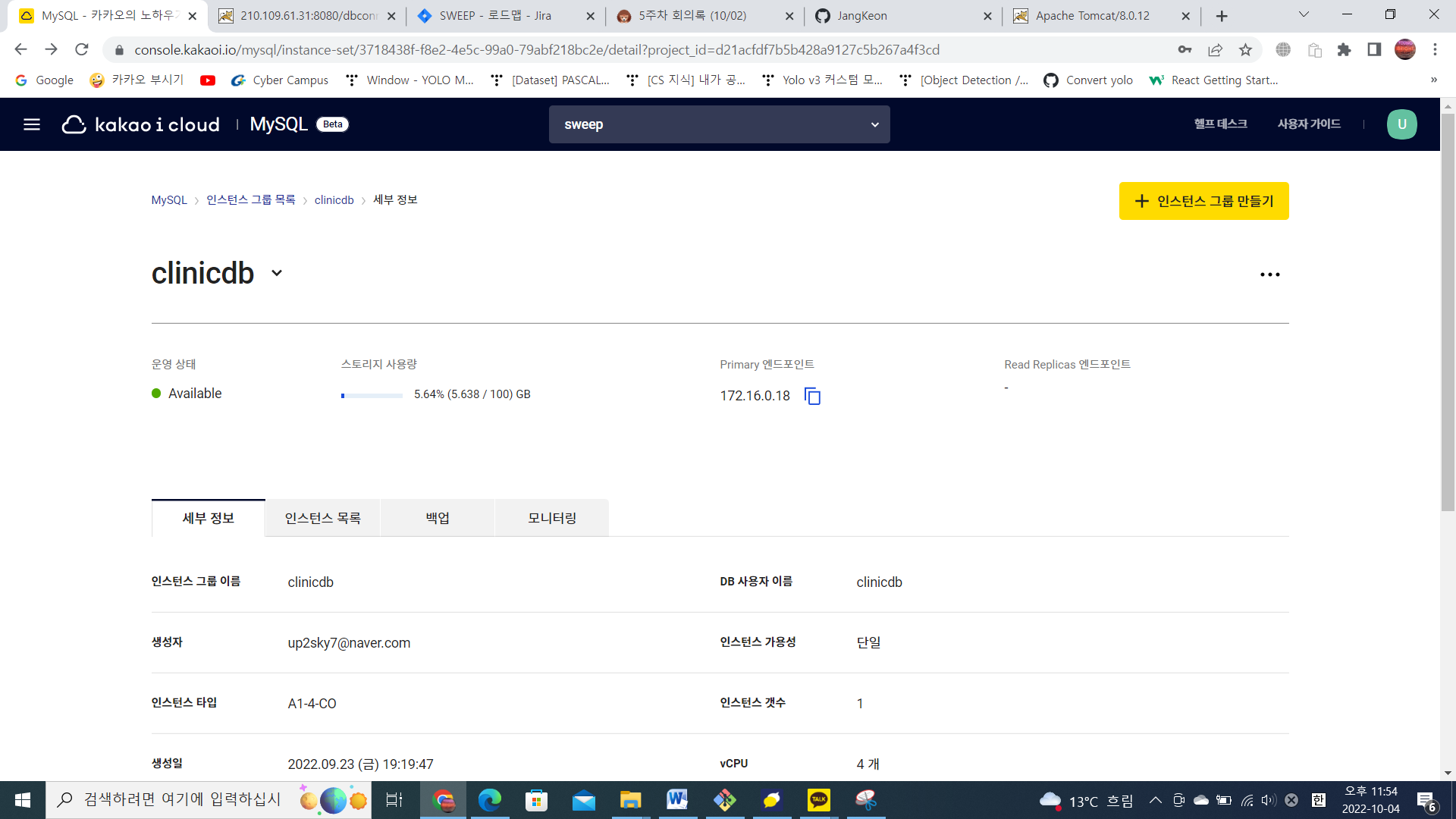
Tomcat 실행 확인

<http://공인아이피:8080> 접속



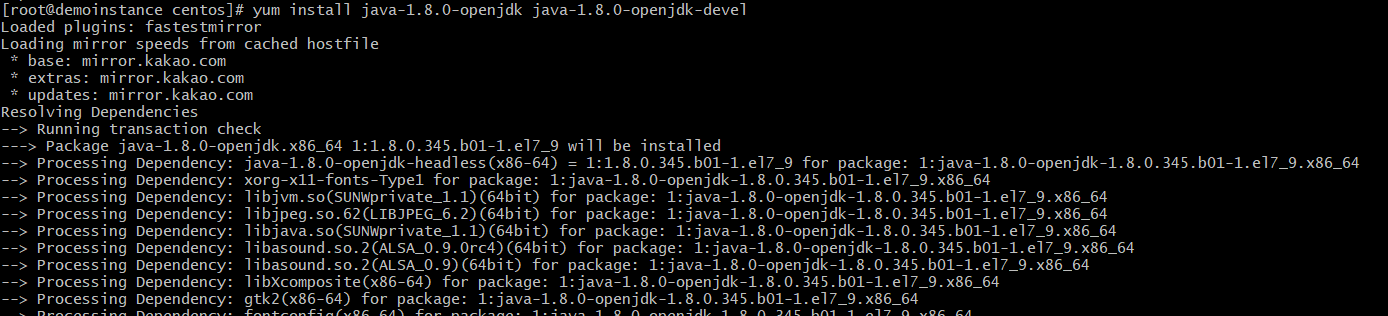
**5. DB Instance 준비하기 (Kakao I Cloud에서 MySQL Instance 생성)**

[[KiC Hands-on] 1. VM 기반의 웹 서비스 구성하기 (oopy.io)](https://kic-gachon.oopy.io/f156508a-b53c-4bcf-be2c-4072fac8efdf) – Step3 따라하기.



**6. CentOS 7 openjdk 1.8 설치**

Yum install java-1.8.0-openjdk java-1.8.0-openjdk-devel

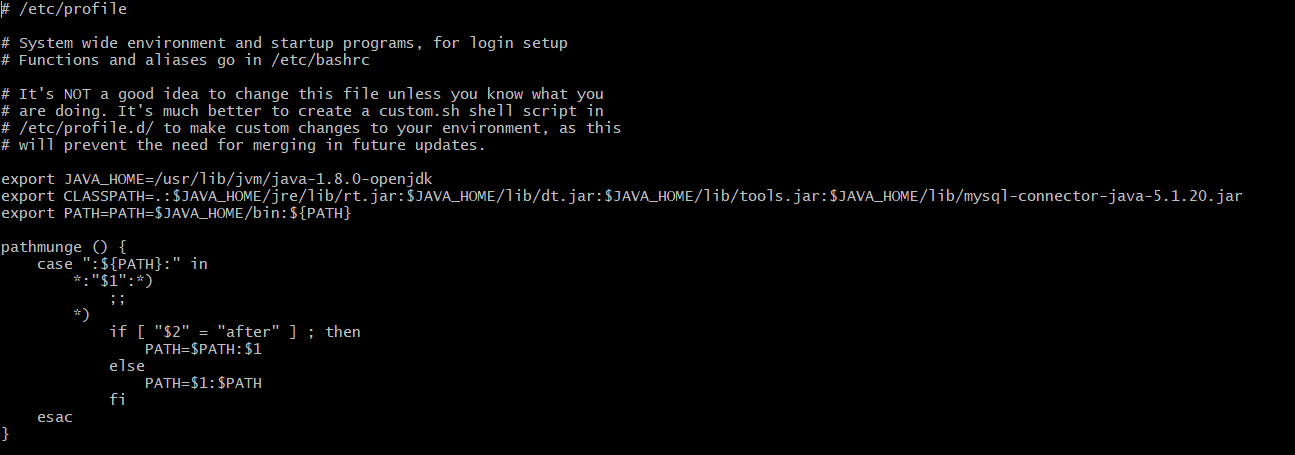


Vi /etc/profile 에 다음 항목들 Path에 맞춰 추가.

export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk

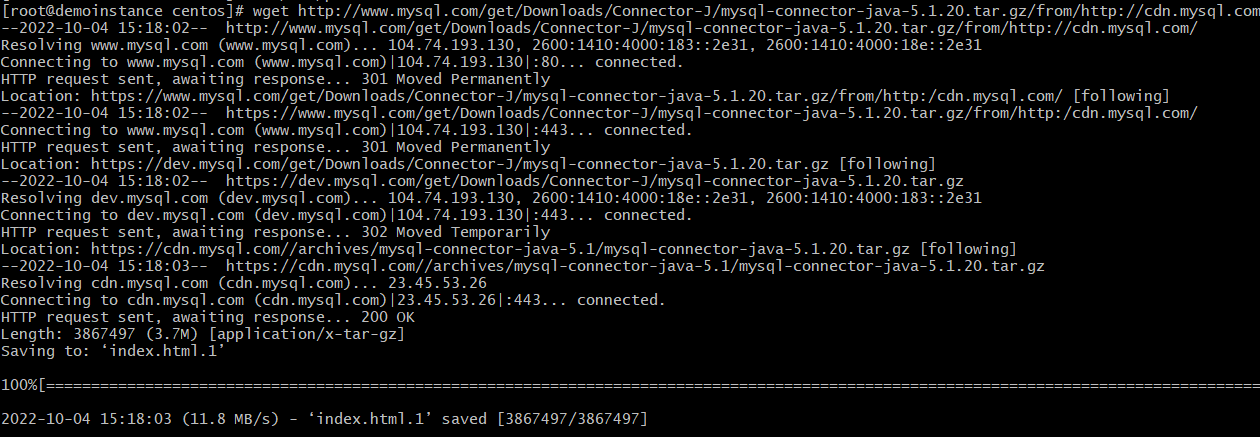
export CLASSPATH=.:$JAVA\_HOME/jre/lib/rt.jar:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$JAVA\_HOME/lib/mysql-connector-java-5.1.47-bin.jar

export PATH=$JAVA\_HOME/bin:${PATH}



**7. MySQL connector 설치, JDBC 연동**

Wget  <http://www.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.20.tar.gz/from/http://cdn.mysql.com/>

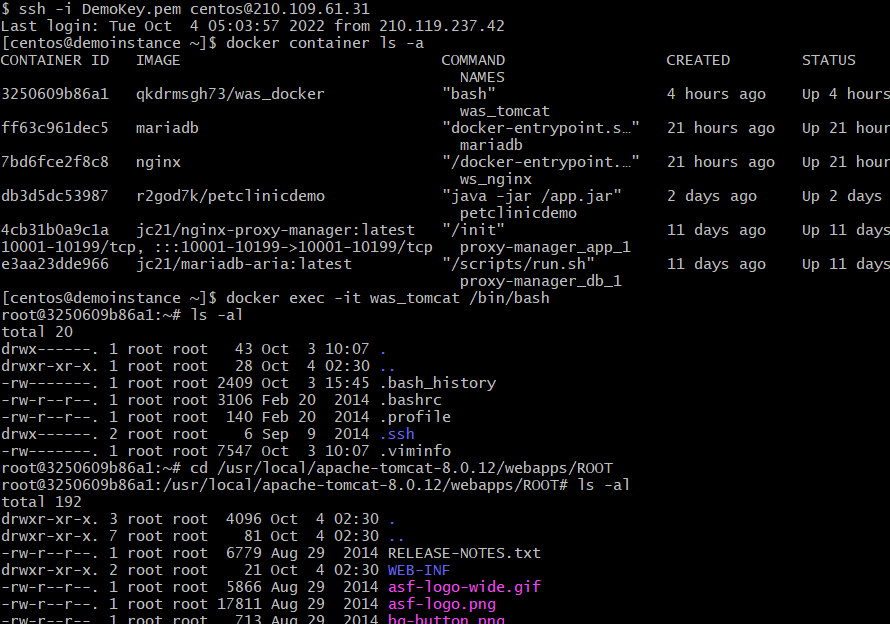


Tar xzvf mysql-connector-java-5.1.20.tar.gz

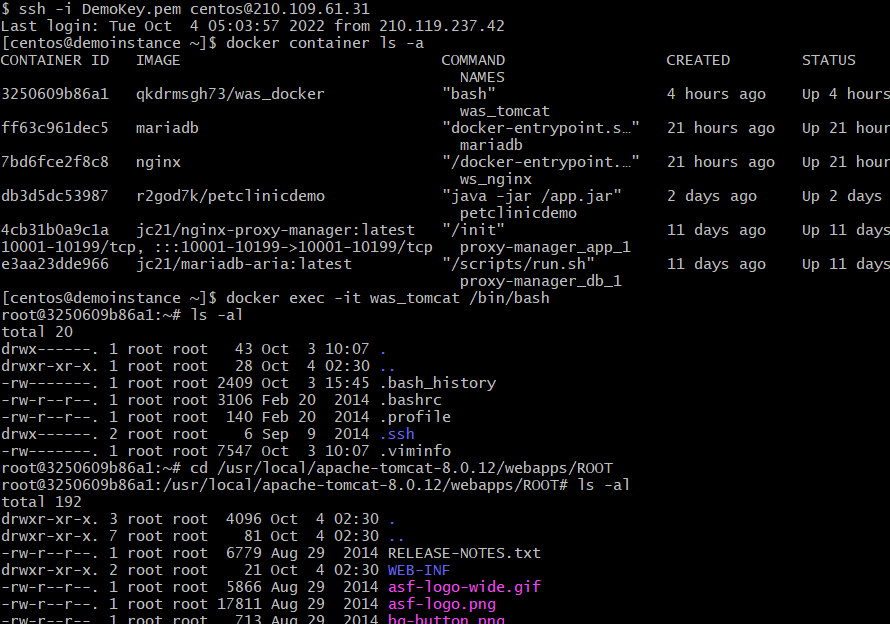
Mv {mysql-connect.jar 파일 이름} /usr/local/apache-tomcat-8.0.12/lib/ (직접 톰캣의 라이브러리 폴더로 옮겨준다.)

**8. Tomcat 컨테이너 접근**

Docker exec –it was\_tomcat /bin/bash



Cd /usr/local/apache-tomcat-8.0.12/webapps/ROOT



DB연동을 위해 dbconnect.jsp 파일을 만들어준다.

Vi dbconnect.jsp

<%@ page import="java.sql.\*" contentType="text/html;charset=utf-8"%>

<%

**String DB\_URL = "jdbc:mysql://** **172.16.0.18/clinicdb";**

**String DB\_USER = "clinicdb";**

**String DB\_PASSWORD= "clinicdb";**

ResultSet rs = null;

Connection conn;

Statement stmt;

try {

Class.forName("com.mysql.jdbc.Driver");

conn = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD);

stmt = conn.createStatement();

// conn.close();

out.println("MariaDB connect : ok");

}catch(Exception e){

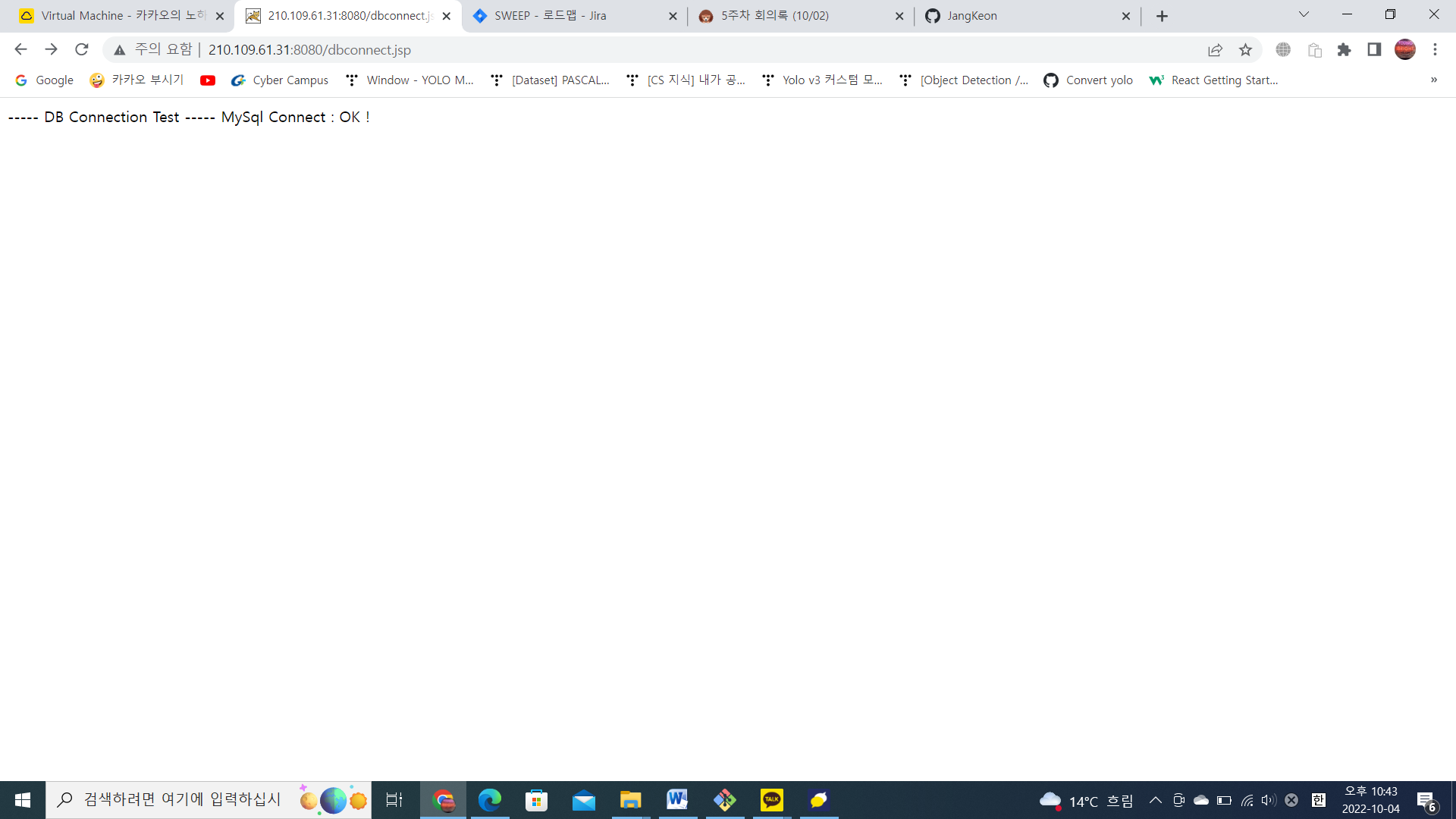
out.println(e);

}

%>

**DB\_URL, USER, PASSWORD**를 본인의 Kakao I Cloud MySQL 인스턴스에 맞게 바꿔서 넣어준다.

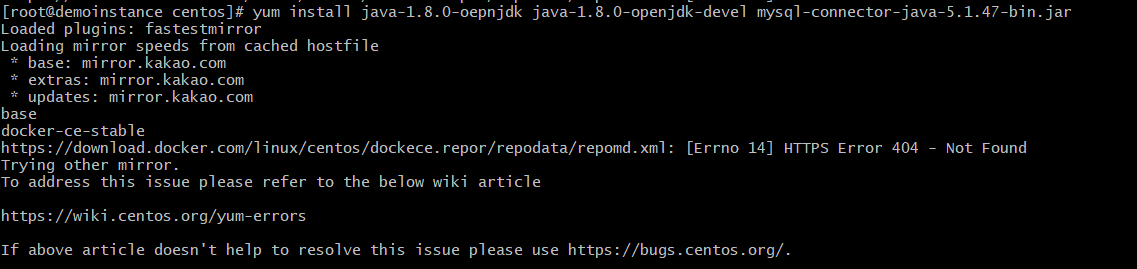
**- 결과 확인**

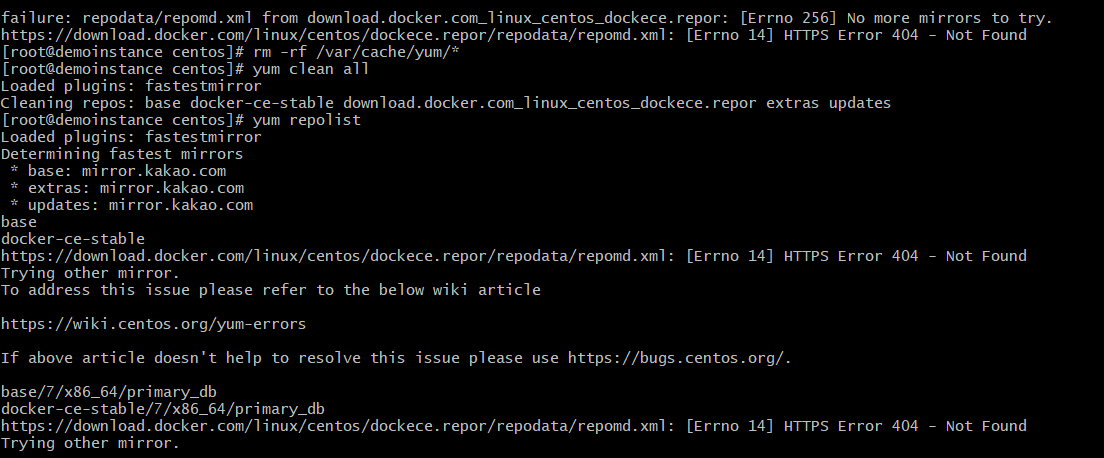


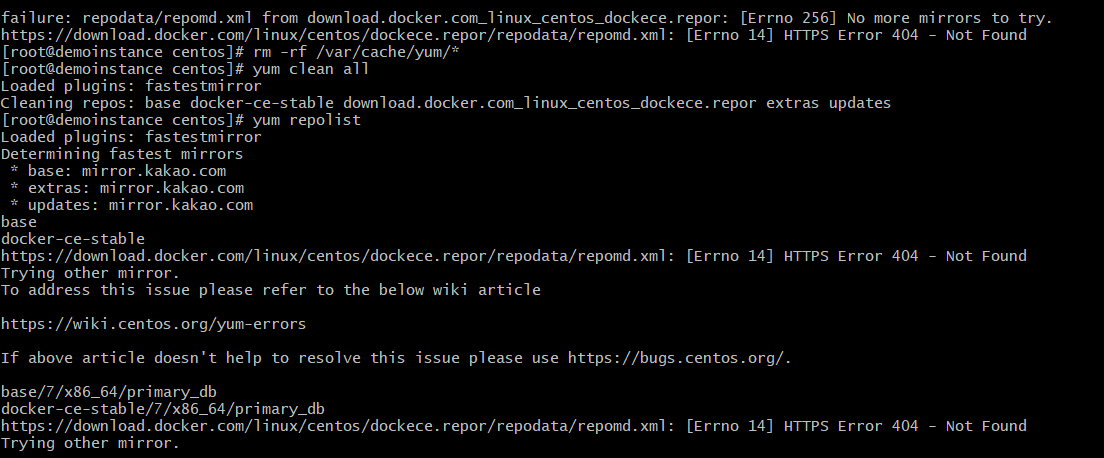
**- Error Handling 1**

3번 과정에서 사이트 주소를 틀리게 적음. https://download.docker.com/linux/centos/docker-ce.repor

위와 같이 적었더니 이후 진행했던 openjdk 와 mysql-connector를 install할 때 [No more mirrors to try] 와 [HTTPS Not Found] 에러가 계속 뜨며 install 되지 않았음. 또한 yum ~ 명령이 모두 실행되지 않았음.

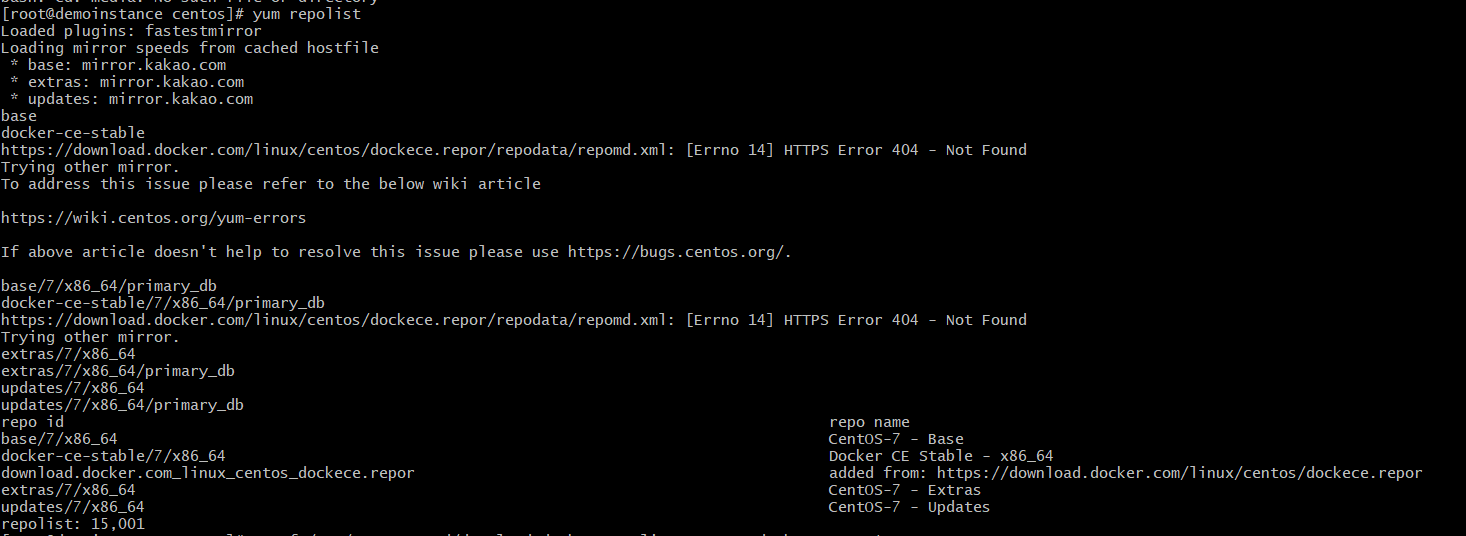






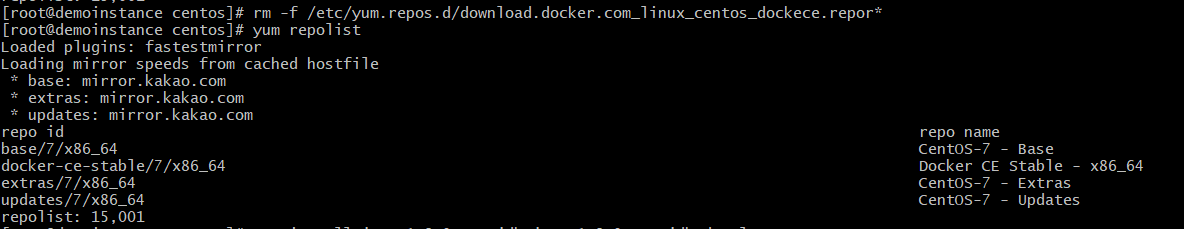
원인은 yum의 repolist에 docker repo가 repor 이라는 이름으로 잘못 추가되어 docker repo를 계속 불러오지 못했던 것.

yum repolist



따라서 잘못 추가되어 있는 repor 를 지우고 다시 repo를 추가해주는 과정을 진행.

Rm –f /etc/yum.repos.d/download.docker.com\_linux\_centos\_dockece.repor\*



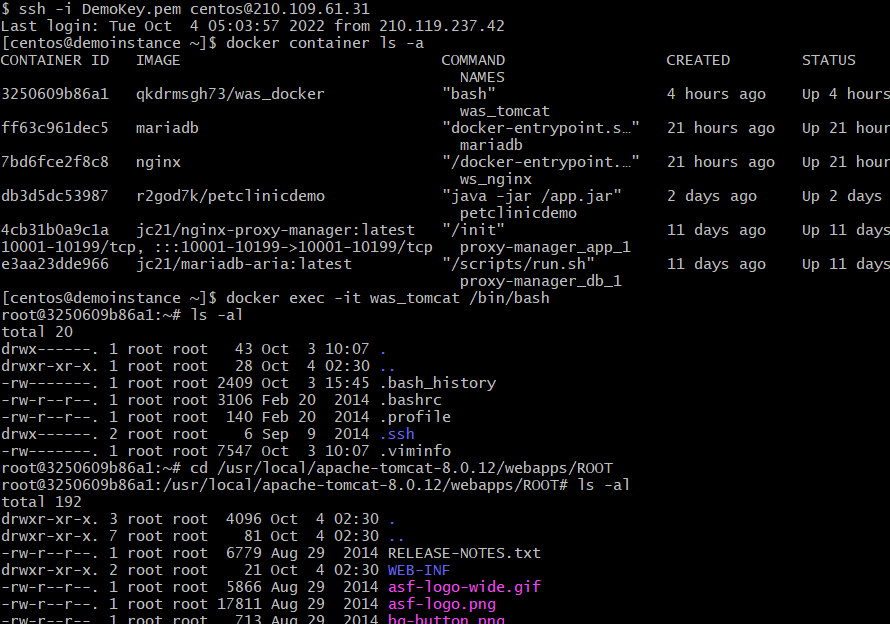
**- Error Handling 2**

Mysql Connector 연결하는 부분에서 파일이 제대로 다운받아 지지 않거나 (tar 파일이 아닌 .html 파일로 변환되어 받아지는) 경우가 많아 새로운 방식으로도 접근해보았음.

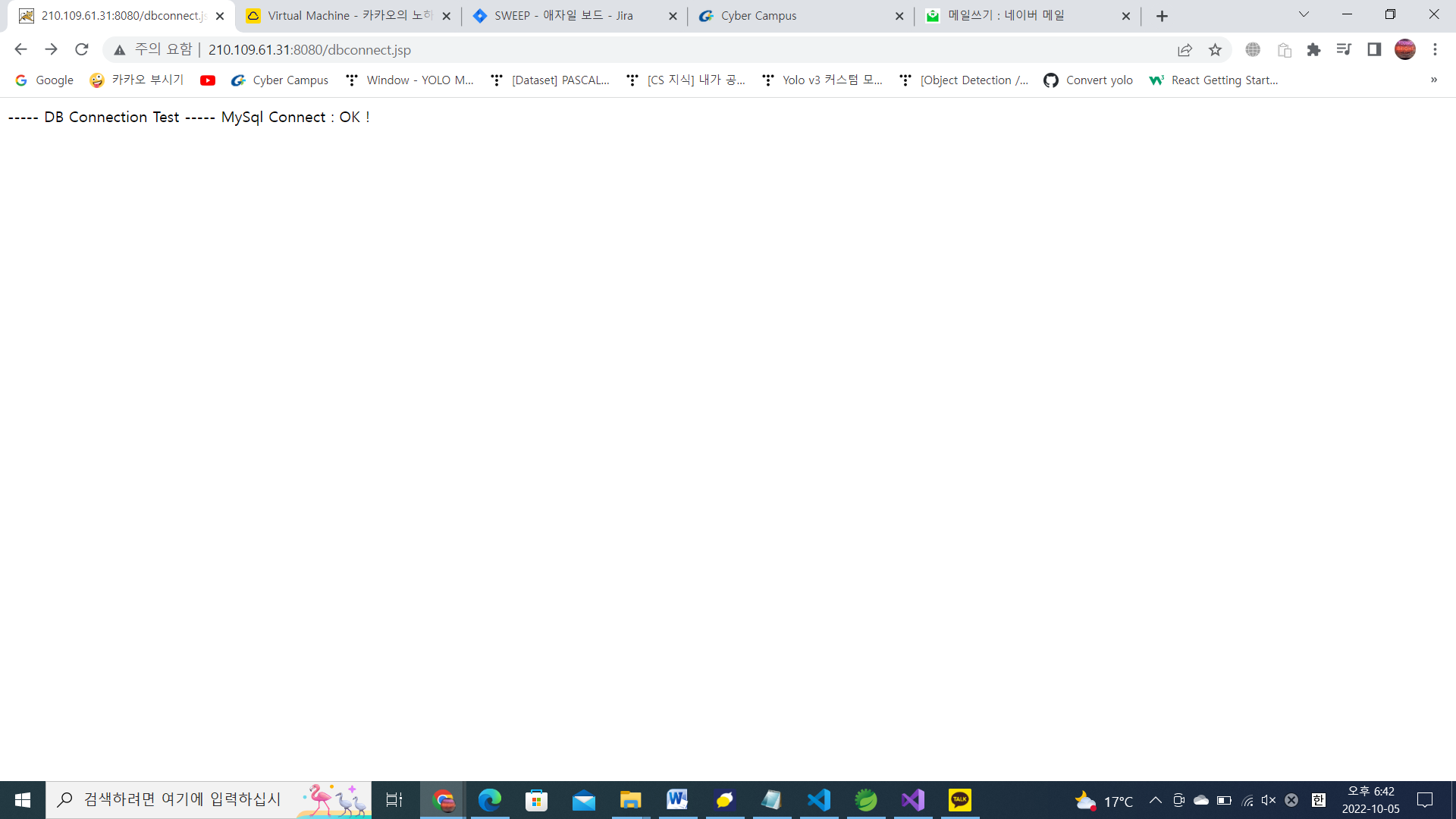
Mysql Connector까지 다운되어 있는 Tomcat 이미지를 Pull 받아서 진행해봄.

Docker pull qkdrmsgh73/was\_docker

Docker run –d –I –t –name was\_tomcat –p 8080:8080 qkdrmsgh73/was\_docker



이후 과정은 위의 4번 WAS 구축 과정 + 8번 Tomcat 컨테이너 접근과 동일.



결과 화면 출력되는 것 확인.

**\* 참고 자료**

리눅스 기반 1Tier WS-WAS-DB 시스템 구축 가이드라인

[[KiC Hands-on] 1. VM 기반의 웹 서비스 구성하기 (oopy.io)](https://kic-gachon.oopy.io/f156508a-b53c-4bcf-be2c-4072fac8efdf)

<https://intrepidgeeks.com/tutorial/configure-centos7-openjdk-18-and-mysql-connector-installation>

<https://claire-song-1995.tistory.com/11>

<https://coding-factory.tistory.com/718>

<https://medium.com/@kimddub/jsp-jsp%EC%99%80-db-%EC%97%B0%EB%8F%99%EC%9C%BC%EB%A1%9C-%EB%B8%94%EB%A1%9C%EA%B7%B8-%EA%B5%AC%ED%98%84%ED%95%B4%EB%B3%B4%EA%B8%B0-73b8ec1df6ea>