C207팀의 TodayFilm 백엔드 서버 배포 과정

Spring Boot를 Maven프로젝트로 생성한뒤 WAR로 빌드하여 배포를 진행하였습니다.
Spring Boot의 자바 버전은 11입니다.
WAR파일 빌드 과정은 생략합니다.

1. 환경 설정
WAR파일로 서버 배포를 진행하기 위해서는 해당 서버에 tomcat과 java가 설치되어 있어야 합니다.
[리눅스 서버 환경]
tomcat 버전 : tomcat9
java 버전 : 1.8.0_341

1_1_1. java 설치
//해당 경로에서 원하는 java 버전을 설치합니다.
https://www.oracle.com/java/technologies/downloads/#java8

1_1_2. 리눅스에서 java 압축 풀기
sudo tar xvf/usr/local/java/jdk-8u341-linux-x64.tar.gz

1_1_3. java 환경 설정 (현재 명령어는 자바가 usr/local/java에 압축이 풀려있어야 합니다.)

```
sudo vi /etc/profile
//환경 변수 설정
export JAVA_HOME=$(readlink -f /usr/local/java |sed“s:bin/java::”)
//java 기본 명령어 등록
sudo update-alternatives --install “/usr/bin/java” “java” “/usr/local/java/jdk1.8.0_341/bin/java” 1;
sudo update-alternatives --install “/usr/bin/javac” “javac” “/usr/local/java/jdk1.8.0_341/bin/javac” 1;
sudo update-alternatives --install “/usr/bin/javaws” “javaws” “/usr/local/java/jdk1.8.0_341/bin/javaaws” 1;

sudo update-alternatives --set java /usr/local/java/jdk1.8.0_341/bin/java;
sudo update-alternatives --set javac /usr/local/java/jdk1.8.0_341/bin/javac;
sudo update-alternatives --set javaws /usr/local/java/jdk1.8.0_341/bin/javaws;

//java 버전 확인
java -version
```

1_2_1. tomcat 설치

sudo apt update

sudo apt upgrade

sudo apt install tomcat9 tomcat9-admin

1.2.2. tomcat 서버 포트 확인

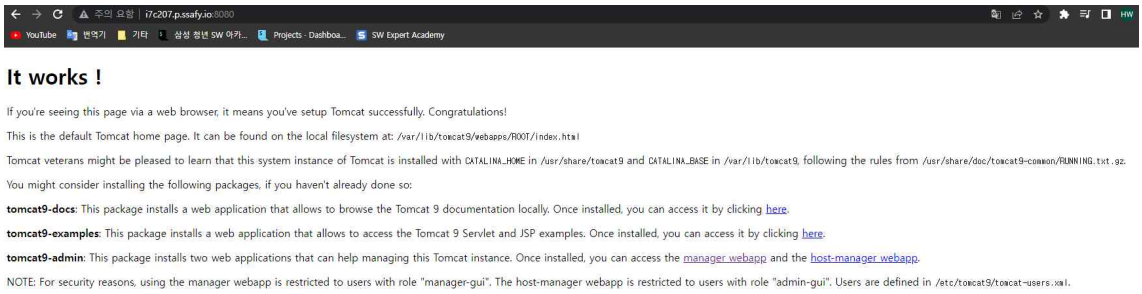
ss -ltn

```
ubuntu@ip-172-26-7-72:/usr/local/java$ ss -ltn
State          Recv-Q         Send-Q                 Local Address:Port                  Peer Address:Port        Process
LISTEN         0              70                         127.0.0.1:33060                    0.0.0.0:*
LISTEN         0              151                          0.0.0.0:3306                     0.0.0.0:*
LISTEN         0              4096                   127.0.0.53%lo:53                       0.0.0.0:*
LISTEN         0              128                          0.0.0.0:22                       0.0.0.0:*
LISTEN         0              128                        127.0.0.1:6010                     0.0.0.0:*
LISTEN         0              100                                *:8080                           *:*
LISTEN         0              128                           [::]:22                          [::]:*
LISTEN         0              128                          [::1]:6010                        [::]:*
```

//tomcat의 기본 서버 포트는 8080입니다.

1.2.3. tomcat 구동 확인

// aws아이피:8080접속한뒤 해당 화면처럼 출력되어 있어야합니다.



1.2.4. tomcat (비)활성

// tomcat9는 서버가 켜지면 자동으로 실행됩니다.

sudo systemctl disable tomcat9 //tomcat 비활성

sudo systemctl enable tomcat9 //tomcat 활성

2. WAR파일 배포

2_1. tomcat 아이디 생성
// tomcat9에서 war파일을 배포하기위해선 tomcat매니저 접속이 가능한 아이디와 패스워드를 생성해야합니다.

//tomcat 아이디 설정
sudo vi /etc/tomcat9/tomcat-users.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<tomcat-users xmlns="http://tomcat.apache.org/xml"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
              version="1.0">
<!--
  NOTE:  By default, no user is included in the "manager-gui" role required
  to operate the "/manager/html" web application.  If you wish to use this app,
  you must define such a user - the username and password are arbitrary. It is
  strongly recommended that you do NOT use one of the users in the commented out
  section below since they are intended for use with the examples web
  application.
-->
<!--
  NOTE:  The sample user and role entries below are intended for use with the
  examples web application. They are wrapped in a comment and thus are ignored
  when reading this file. If you wish to configure these users for use with the
  examples web application, do not forget to remove the <!.. ..> that surrounds
  them. You will also need to set the passwords to something appropriate.
-->
  <role rolename="tomcat"/>
  <role rolename="role1"/>
  <user username="tomcat" password="[redacted]" roles="admin-gui,manager-gui"/>
<!--
  <user username="both" password="<must-be-changed>" roles="tomcat,role1"/>
  <user username="role1" password="<must-be-changed>" roles="role1"/>
-->
</tomcat-users>
```

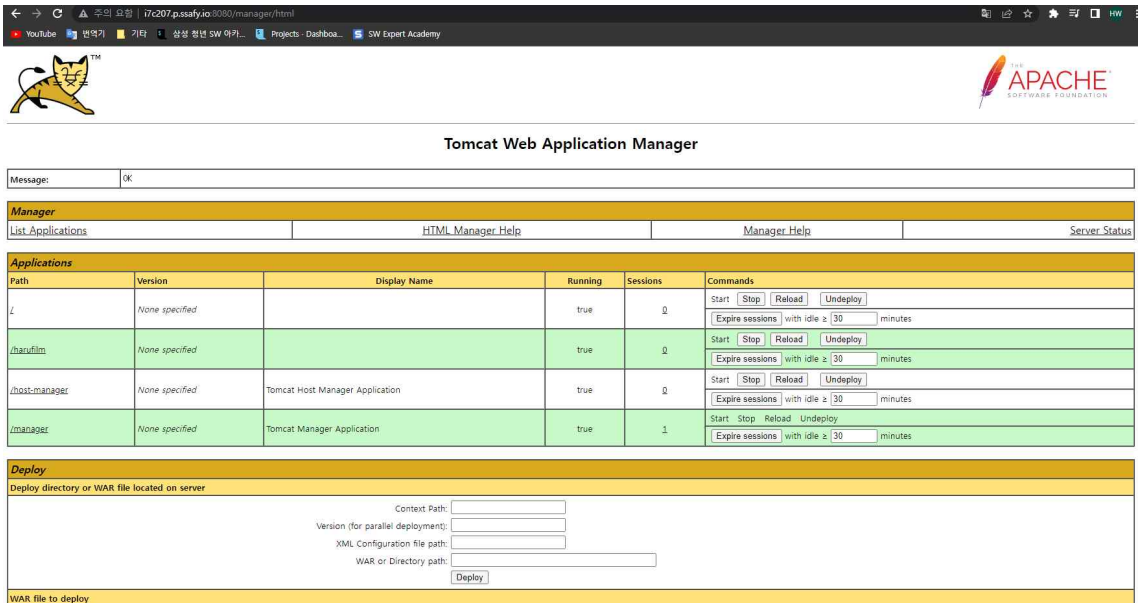
//해당 위치의 주석을 풀고 화면 같이 입력해줍니다. (password 입력)

sudo systemctl restart tomcat9
//톰캣을 재시작 해줍니다.

2_2. tomcat 매니저 접속

// aws아이피:8080/manager/html 해당URL로 접속한뒤 방금 작성한 사용자이름,비밀번호를 입력 해줍니다.

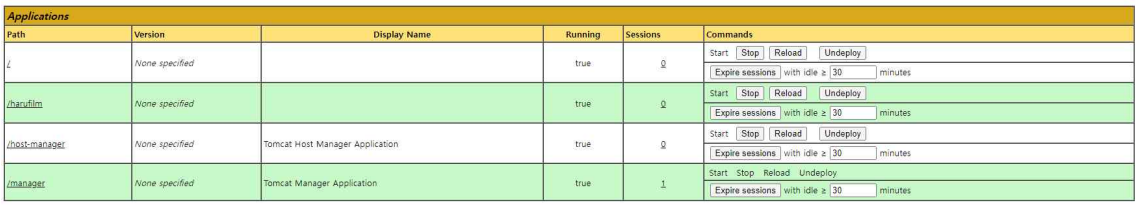
//결과 화면



2_3. war파일 배포

// WAR file to deploy에 파일 선택을 클릭하여 빌드한 WAR파일을 선택한뒤 Deploy버튼을 클릭해줍니다.

2_4. 배포 확인



// 예시로 harufilm이 배포가 되어 있습니다.
배포한 war파일을 접근하는 방법은
aws아이피:8080/harufilm(war파일이름)/api 으로 접근합니다.

3. 파일 저장

프론트에서 파일을 전송할 때 서버에 파일을 저장하기 위해서는 추가적인 설정이 필요합니다.

3_1. Spring boot CORS 설정

```java
package com.ssafy.harufilm.config;

import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.config.annotation.CorsRegistry;
import org.springframework.web.servlet.config.annotation.ResourceHandlerRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

// CORS 설정
@Configuration
public class WebConfig implements WebMvcConfigurer {

    private String connectPath = "/upload/**";
    private String resourcePath = "file:/var/opt/upload/";

    @Override
    public void addResourceHandlers(ResourceHandlerRegistry registry) {
        registry.addResourceHandler(connectPath)
                .addResourceLocations(resourcePath);

    }

    @Override
    public void addCorsMappings(CorsRegistry registry) {
        registry.addMapping(pathPattern: "/**")
                .allowedOrigins(...origins: "*")
                .allowedMethods(...methods: "*");
    }
}
```

Spring boot코드에서 파일 접근이 가능한 경로를 설정해줍니다.

3_2. tomcat 파일 저장 경로 설정

sudo vi /etc/systemd/system/multi-user.target.wants/tomcat9.service

```
#
# Systemd unit file for Apache Tomcat
#

[Unit]
Description=Apache Tomcat 9 Web Application Server
Documentation=https://tomcat.apache.org/tomcat-9.0-doc/index.html
After=network.target
RequiresMountsFor=/var/log/tomcat9 /var/lib/tomcat9

[Service]

# Configuration
Environment="CATALINA_HOME=/usr/share/tomcat9"
Environment="CATALINA_BASE=/var/lib/tomcat9"
Environment="CATALINA_TMPDIR=/tmp"
Environment="JAVA_OPTS=-Djava.awt.headless=true"

# Lifecycle
Type=simple
ExecStartPre=+/usr/libexec/tomcat9/tomcat-update-policy.sh
ExecStart=/bin/sh /usr/libexec/tomcat9/tomcat-start.sh
SuccessExitStatus=143
Restart=on-abort

# Logging
SyslogIdentifier=tomcat9

# Security
User=tomcat
Group=tomcat
PrivateTmp=yes
AmbientCapabilities=CAP_NET_BIND_SERVICE
NoNewPrivileges=true
CacheDirectory=tomcat9
CacheDirectoryMode=750
ProtectSystem=strict
ReadWritePaths=/etc/tomcat9/Catalina/
ReadWritePaths=/var/lib/tomcat9/webapps/
ReadWritePaths=/var/log/tomcat9/
ReadWritePaths=/var/opt/upload/


[Install]
WantedBy=multi-user.target
~
```

// 해당 위치에 Springboot에 작성한 경로를 작성해줍니다.

3_3. 파일 경로 권한 설정

sudo chown -R tomcat:tomcat /var/opt/upload

sudo chmod 777 /var/opt/upload

3_4. tomcat 재시작

sudo systemctl daemon-reload //units 리로드

sudo systemctl restart tomcat9