





Profile



서동윤 1997.2.12

 충청남도 천안시 서북구 두정동

 sdy2766@naver.com / 010-9105-2766

 <https://github.com/dongyunseo?tab=repositories>

2012.3 ~ 2015.2	천안상업고등학교 입학 천안상업고등학교 졸업	2015.3 ~ 2020.2	백석대학교 정보통신학부 입학 백석대학교 정보통신학부 졸업
2020.7~ 2020.12	휴먼교육센터 웹 개발 교육 이수	2023.8~ 2024.1	빅데이터 분석과정 이수
2021.1~ 2021.12	에스알포스트/ 운영지원팀	2022.1~ 2023.8	태강정보통신/ 운영지원팀

명칭 : 플랫폼 별 음원 차트 서비스

개발 기간 : 2024.2. ~ 2024.4. (2개월)

플랫폼 : Web

개발 인원 : 1명 (개인 프로젝트)

개발 환경

언어 : Java (JDK 1.8), HTML, CSS, JavaScript, Python

서버 : Apache Tomcat 8.5

프레임워크 : Spring Mvc2, Dynamic Web

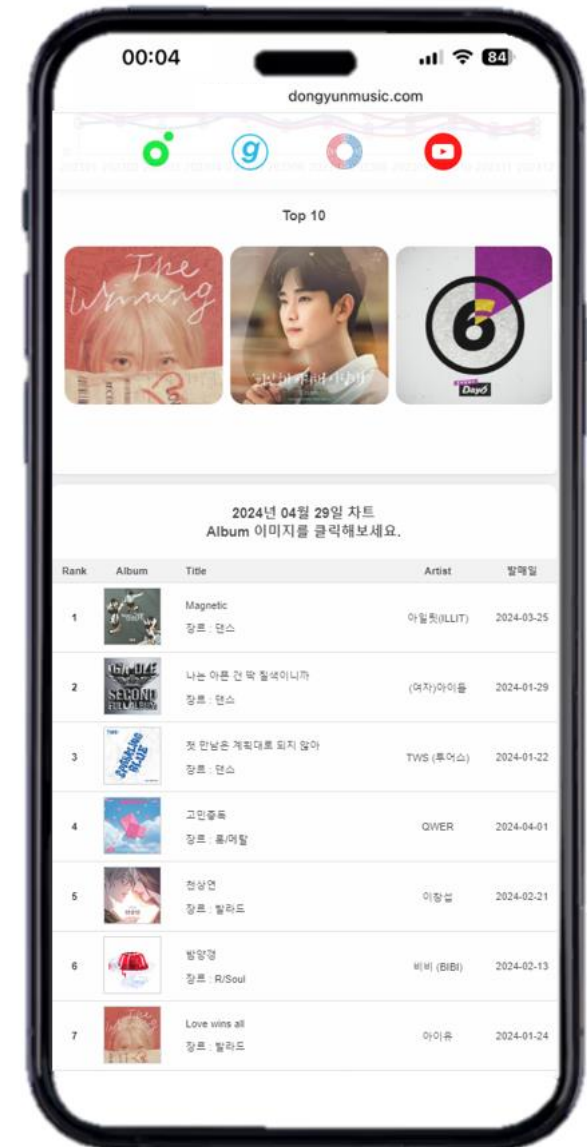
DB : Oracle 11g, MySQL

Server : Apache tomcat 8.5, AWS EC2, Ubuntu

Tool : STS-3, Putty, Git, Visual Studio Code

웹 호스팅 URL

http://www.dongyunmusic.com:8080/Music/Main_Page/Main.dk



WBS

[illegible]

프로젝트 설명

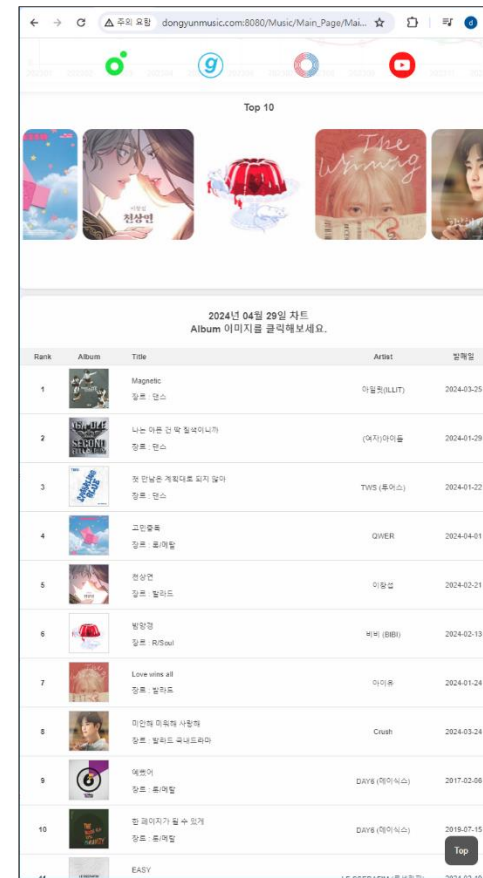
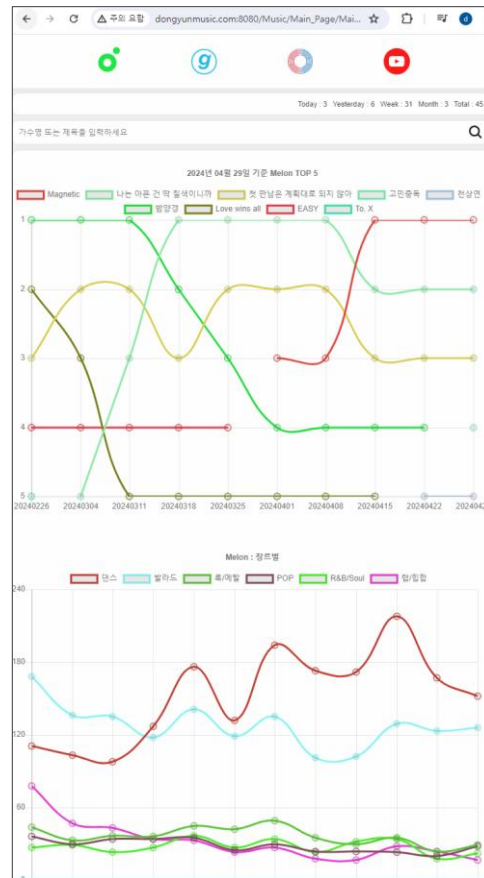
2. 플랫폼 별 음원 차트 서비스

2.1 웹 접속 시 IP주소를 가져와 ViewCount를 계산합니다.

2.2 주단위로 플랫폼 별 TOP5음원에 대한 그래프를 확인할 수 있습니다.

2.3 전년도 월별, 장르별 그래프를 확인할 수 있습니다.

2.4 검색기능을 통하여 가수, 제목으로 유사곡을 검색할 수 있습니다.



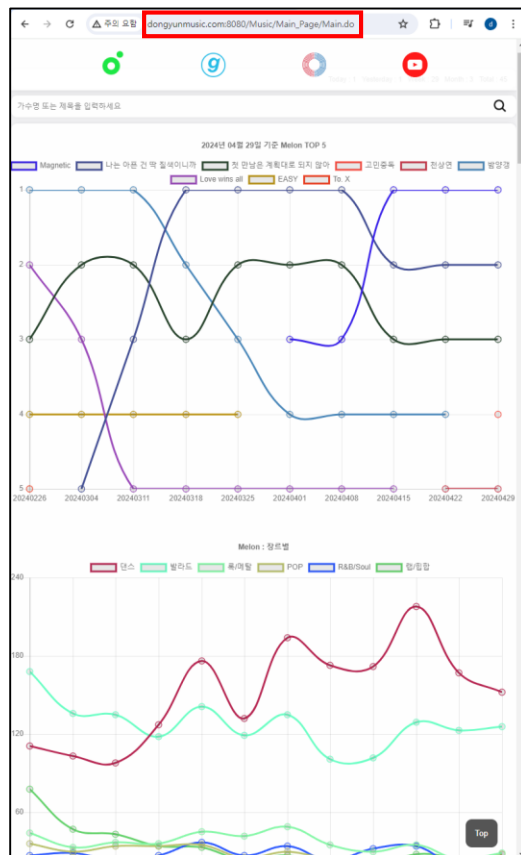
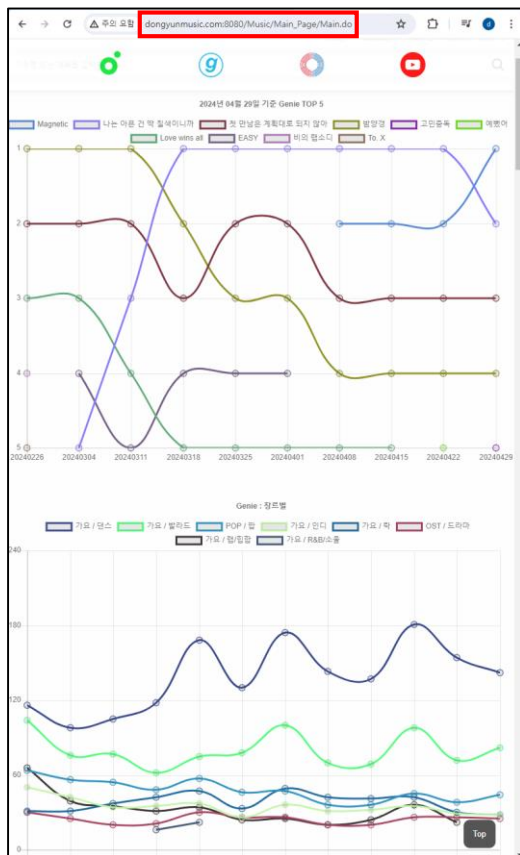
The screenshot displays the website's interface for the search results page. The table shows the search results for the query '가수명 또는 제목을 입력하세요'.

Number	img	title	Singer
1		사랑하지 않아서 그뿐니	지아
2		사랑하지 않아서 그뿐니	지아
3		사랑하지 않아서 그뿐니	지아
4		사랑하지 않아서 그뿐니	지아
5		사랑하지 않아서 그뿐니	지아
6		사랑하지 않아서 그뿐니	지아
7		사랑하지 않아서 그뿐니	지아
8		사랑하지 않아서 그뿐니	지아
9		사랑하지 않아서 그뿐니	지아
10		사랑하지 않아서 그뿐니	지아

2. 플랫폼 별 음원 차트 서비스

2.5 화면의 그래프나 TOP100 자료는 AJAX를 통해 데이터를 교환하므로

페이지 이동이 없으며 서버 부담이 줄어들어 빠르게 검색할 수 있습니다



```

Main.jsp
1 var imageUrl // 전역 범위에 imageUrl 변수를 정의합니다
2
3 $(document).ready(function(){
4   $(".nav-link").click(function(){
5     var buttonId = $(this).attr("id"); // 클릭된 버튼의 ID
6     $.ajax({
7       type: "POST",
8       url: "/AjaxTop100Servlet",
9       data: { Platform: buttonId }, // 클릭된 버튼의 ID를 전달
10      success: function(response) {
11        var urlArray = response.split(",");
12        var newDate_List = [];
13        for (var i = 0; i < urlArray.length; i += 9) { // 9개씩
14          var releaseDate = new Date(urlArray[i + 6]);
15
16          newDate_List.push({
17            "Day": urlArray[i],
18            "Rank": parseInt(urlArray[i + 1]),
19            "Title": urlArray[i + 2].replace(/\\/g, ""),
20            "Singer": urlArray[i + 3],
21            "album_img_urls": urlArray[i + 4],
22            "release_date": urlArray[i + 5],
23            "genre_text": urlArray[i + 6],
24            "View_Count": urlArray[i + 7],
25            "click_Url": urlArray[i + 8]
26          }); // 객체 생성하여 newDate에 추가
27
28          appendTop100(newDate_List, buttonId); // 이마
29        }
30      }
31    });
32    return false; // 기본 동작 중지
33  });
34
35  // 조회수 천 단위, 불이는 스크립트
36  function addCommasToNumber(number) {
37    return number.toString().replace(/B(?=(\d{3})+)?/g, ",");
38  }
39
Genre_line_chart.js
1 $(document).ready(function(){
2   var lineChart2;
3   if (typeof lineChart2 !== 'undefined') {
4     lineChart2.destroy();
5     $("#line-chart2").html(""); // line-chart2 요소 내용을 비웁니다
6   }
7   $(".nav-link").click(function(){
8     var buttonId = $(this).attr("id"); // 클릭된 버튼의 ID를 가져
9     // 이전 차트가 있다면 삭제합니다.
10    if (typeof lineChart2 !== 'undefined') {
11      lineChart2.destroy();
12    }
13    $.ajax({
14      type: "POST",
15      url: "/GenreAjaxServlet", // 두 번째 DAO의 엔드포인트 URL
16      data: { Platform: buttonId }, // 클릭된 버튼의 ID를 전달
17      success: function(response) {
18        var data = response.split(","); // 받은 데이터를 쉼표와 공
19        var newDate = []; // 배열로 선언
20
21        for (var i = 0; i < data.length; i += 3) { // 받은 데이터를
22          newDate.push({ "Day": data[i], "Rank": parseInt(data[i + 1]) });
23        }
24
25        // 차트 데이터 생성
26        var chartData = [];
27
28        // newDate 배열에서 중복을 제거한 Day 값만 추출하여
29        var uniqueDays = [...new Set(newDate.map(entry => entry.Day))];
30
31        // 중복 제거된 Day 값을 오름차순으로 정렬 (uniqueDays)
32        uniqueDays.sort((a, b) => a.localeCompare(b, undefined));
33
34        // 최신 날짜 추출
35        var maxRank = Math.max(...uniqueDays);
36        // 문자열로 변환
37        var dateString = maxRank.toString();
38        // 연도, 월, 일 추출
39        var year = dateString.substr(0, 4);
40        var month = dateString.substr(4, 2);
41        var day = dateString.substr(6, 2);
42        // 날짜 형식으로 변환
43        var formattedDate = year + "년 " + month + "월 " + day + "일";
44
45        // x축의 값
46        lineChart = new Chart(document.getElementById("line-chart2"), {
47          type: 'line',
48          data: {
49            // Day 값을 groupBy해서 놓기
50

```

2. 플랫폼 별 음원 차트 서비스

2.6 Top 100을 추출하는 SQL문은 항상 최신 날짜에 해당하는 데이터를 가져옵니다.

30 • SELECT Day, Ranking, Title, Singer, album_img_urls, release_date, genre_text, view_Count, click_Url FROM Music WHERE 1=1 AND Platform = 'Melon' and Day = (Select max(Day) from Music where Platform = 'Melon') order by Ranking asc LIMIT 100

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Fetch rows:

	Day	Ranking	Title	Singer	album_img_urls	release_date	genre_text	view_Count	click_Url
▶	20240429	1	Magnetic	아일릿(ILLIT)	https://cdnimg.melon.co.kr/cm2/album/images/114/50/069/11450069_20240325110702_500.jpg/melon/resize/240/quality/160/optimize	2024-03-25	댄스	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	2	나는 아픈 건 딱 질색이니까	(여자)아이들	https://cdnimg.melon.co.kr/cm2/album/images/114/02/655/11402655_20240129121016_500.jpg/melon/resize/240/quality/160/optimize	2024-01-29	댄스	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	3	첫 만남은 계획대로 되지 않아	TWS (투어스)	https://cdnimg.melon.co.kr/cm2/album/images/113/91/902/11391902_20240122132041_500.jpg/melon/resize/240/quality/160/optimize	2024-01-22	댄스	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	4	고민종독	QWER	https://cdnimg.melon.co.kr/cm2/album/images/114/54/681/11454681_20240328174504_500.jpg/melon/resize/240/quality/160/optimize	2024-04-01	록/메탈	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	5	천상연	이창섭	https://cdnimg.melon.co.kr/cm2/album/images/114/21/941/11421941_20240220105008_500.jpg/melon/resize/240/quality/160/optimize	2024-02-21	발라드	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	6	방안경	비비 (BIBI)	https://cdnimg.melon.co.kr/cm2/album/images/114/15/997/11415997_20240226162412_500.jpg/melon/resize/240/quality/160/optimize	2024-02-13	R/Soul	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	7	Love wins all	아이유	https://cdnimg.melon.co.kr/cm2/album/images/114/04/142/11404142_20240220141548_500.jpg/melon/resize/240/quality/160/optimize	2024-01-24	발라드	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	8	미안해 미워해 사랑해	Crush	https://cdnimg.melon.co.kr/cm2/album/images/114/49/278/11449278_20240322170237_500.jpg/melon/resize/240/quality/160/optimize	2024-03-24	발라드 국내드라마	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	9	예뻐어	DAY6 (데이식스)	https://cdnimg.melon.co.kr/cm/album/images/100/36/099/10036099_500.jpg/melon/resize/240/quality/160/optimize	2017-02-06	록/메탈	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	10	한 페이지가 될 수 있게	DAY6 (데이식스)	https://cdnimg.melon.co.kr/cm/album/images/103/07/346/10307346_500.jpg/melon/resize/240/quality/160/optimize	2019-07-15	록/메탈	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	11	EASY	LE SSERAFIM (...)	https://cdnimg.melon.co.kr/cm2/album/images/114/07/156/11407156_20240219142535_500.jpg/melon/resize/240/quality/160/optimize	2024-02-19	랩/힙합	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	12	비의 랩소디	임재현	https://cdnimg.melon.co.kr/cm2/album/images/113/76/816/11376816_20231201120653_500.jpg/melon/resize/240/quality/160/optimize	2023-12-03	발라드	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	13	Smart	LE SSERAFIM (...)	https://cdnimg.melon.co.kr/cm2/album/images/114/07/156/11407156_20240219142535_500.jpg/melon/resize/240/quality/160/optimize	2024-02-19	R/Soul	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	14	To. X	태연 (TAEYEON)	https://cdnimg.melon.co.kr/cm2/album/images/113/72/896/11372896_20231127104957_500.jpg/melon/resize/240/quality/160/optimize	2023-11-27	R/Soul	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	15	SPOT! (feat. JENNIE)	지코 (ZICO)	https://cdnimg.melon.co.kr/cm2/album/images/114/74/894/11474894_20240426103349_500.jpg/melon/resize/240/quality/160/optimize	2024-04-26	랩/힙합	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	16	그대만 있다면 (여름날 우리 ...	너드커넥트 (Ne...	https://cdnimg.melon.co.kr/cm2/album/images/113/05/759/11305759_20230814111006_500.jpg/melon/resize/240/quality/160/optimize	2023-08-15	발라드	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	17	SHEESH	BABYMONSTER	https://cdnimg.melon.co.kr/cm2/album/images/114/54/802/11454802_20240401174250_500.jpg/melon/resize/240/quality/160/optimize	2024-04-01	R/Soul	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	18	헤어자 말해요	박재정	https://cdnimg.melon.co.kr/cm2/album/images/112/27/533/11227533_20230419162238_500.jpg/melon/resize/240/quality/160/optimize	2023-04-20	발라드	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	19	Perfect Night	LE SSERAFIM (...)	https://cdnimg.melon.co.kr/cm2/album/images/113/52/904/11352904_20231027101104_500.jpg/melon/resize/240/quality/160/optimize	2023-10-27	댄스	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	20	Love 119	RIIZE	https://cdnimg.melon.co.kr/cm2/album/images/113/94/065/11394065_20240104151541_500.jpg/melon/resize/240/quality/160/optimize	2024-01-05	댄스	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	21	에피소드	이무진	https://cdnimg.melon.co.kr/cm2/album/images/113/82/987/11382987_20231212165916_500.jpg/melon/resize/240/quality/160/optimize	2023-12-13	록/메탈	NULL	https://www.melon.com/song/detail.htm?songI...
	20240429	22	I AM	IVE (아이브)	https://cdnimg.melon.co.kr/cm2/album/images/112/11/297/11211297_20230410151046_500.jpg/melon/resize/240/quality/160/optimize	2023-04-10	댄스	NULL	https://www.melon.com/song/detail.htm?songI...

Music 11 ×

2. 플랫폼 별 음원 차트 서비스

2.7 화면 맨 상단의 TOP5 그래프 및 장르별 비교 그래프를 추출하기 위한 SQL문입니다.

```
26 • SELECT Day, Ranking, Title FROM Music
27 WHERE Platform = 'Melon'
28 AND STR_TO_DATE(SUBSTR(Day, 1, 8), '%Y%m%d') >= DATE_SUB((SELECT STR_TO_DATE(MAX(DAY), '%Y%m%d') FROM Music), INTERVAL 63 DAY)
29 AND STR_TO_DATE(SUBSTR(Day, 1, 8), '%Y%m%d') <= DATE_ADD(DATE_SUB((SELECT STR_TO_DATE(MAX(DAY), '%Y%m%d') FROM Music), INTERVAL 7 DAY), INTERVAL 1 WEEK)
30 AND Ranking < 6 ORDER BY Day DESC, Ranking;
31
```

Result Grid | Filter Rows: | Export: | Wrap Cell Contents: |

Day	Ranking	Title
20240429	1	Magnetic
20240429	2	나는 아픈 건 막 질색이니까
20240429	3	첫 만남은 계획대로 되지 않아
20240429	4	고민종류
20240429	5	전상연
20240422	1	Magnetic
20240422	2	나는 아픈 건 막 질색이니까
20240422	3	첫 만남은 계획대로 되지 않아
20240422	4	밤알경
20240422	5	전상연
20240415	1	Magnetic
20240415	2	나는 아픈 건 막 질색이니까
20240415	3	첫 만남은 계획대로 되지 않아
20240415	4	밤알경
20240415	5	Love wins all
20240408	1	나는 아픈 건 막 질색이니까
20240408	2	첫 만남은 계획대로 되지 않아
20240408	3	Magnetic
20240408	4	밤알경
20240408	5	Love wins all
20240401	1	나는 아픈 건 막 질색이니까
20240401	2	첫 만남은 계획대로 되지 않아

Music 7 x

Output

Action Output

#	Time	Action	Message
12	15:44:40	SELECT Day, Ranking, Title FROM Music WHERE Platform = 'Melon' AND STR_TO_DATE(SUBSTR(Day, 1, 8), '%Y%m%d') >= DAT...	50 row(s) returned

```
8 • select substr(Day,1,6) as Day,
9 count(Genre_Text) as Ranking,
10 CASE
11 WHEN Genre_Text = '발라드' OR Genre_Text = '발라드 국내트롯' THEN '발라드'
12 WHEN Genre_Text = '발라드' OR Genre_Text = '발라드 국내영화' THEN '발라드'
13 WHEN Genre_Text = '발라드' OR Genre_Text = '발라드 인디음악' THEN '발라드'
14 ELSE Genre_Text
15 END as Title
16 from Music
17 where 1=1
18 and Platform = 'Melon' and Genre_Text is not null
19 and substr(Day,1,4) = '2023'
20 group by substr(Day,1,6),
21 CASE
22 WHEN Genre_Text = '발라드' OR Genre_Text = '발라드 국내트롯' THEN '발라드'
23 WHEN Genre_Text = '발라드' OR Genre_Text = '발라드 국내영화' THEN '발라드'
24 WHEN Genre_Text = '발라드' OR Genre_Text = '발라드 인디음악' THEN '발라드'
25 ELSE Genre_Text
26 END
27 HAVING count(Genre_Text) >= 16;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Contents: |

Day	Ranking	Title
202312	126	발라드
202312	152	단소
202312	17	합/유희
202312	29	록/에달
202312	22	R&B/Soul
202312	28	POP
202311	167	단소
202311	123	발라드
202311	20	POP
202311	24	합/유희
202311	24	록/에달
202311	18	R&B/Soul
202310	218	단소
202310	129	발라드
202310	28	합/유희
202310	35	록/에달
202310	34	R&B/Soul
202310	23	POP
202309	172	단소
202309	102	발라드
202309	32	R&B/Soul
202309	24	POP

Result 10 x

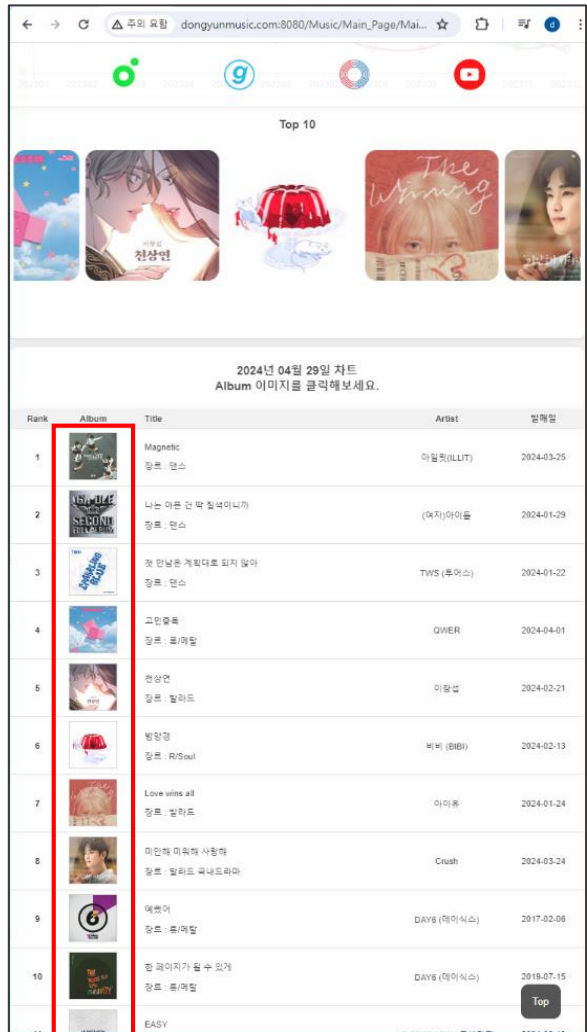
Output

Action Output

#	Time	Action	Message
15	15:45:14	select substr(Day,1,6) as Day, count(Genre_Text) as Ranking, CASE WHEN Genre_Text = '발라드' OR Genre_Text = ...	72 row(s) returned

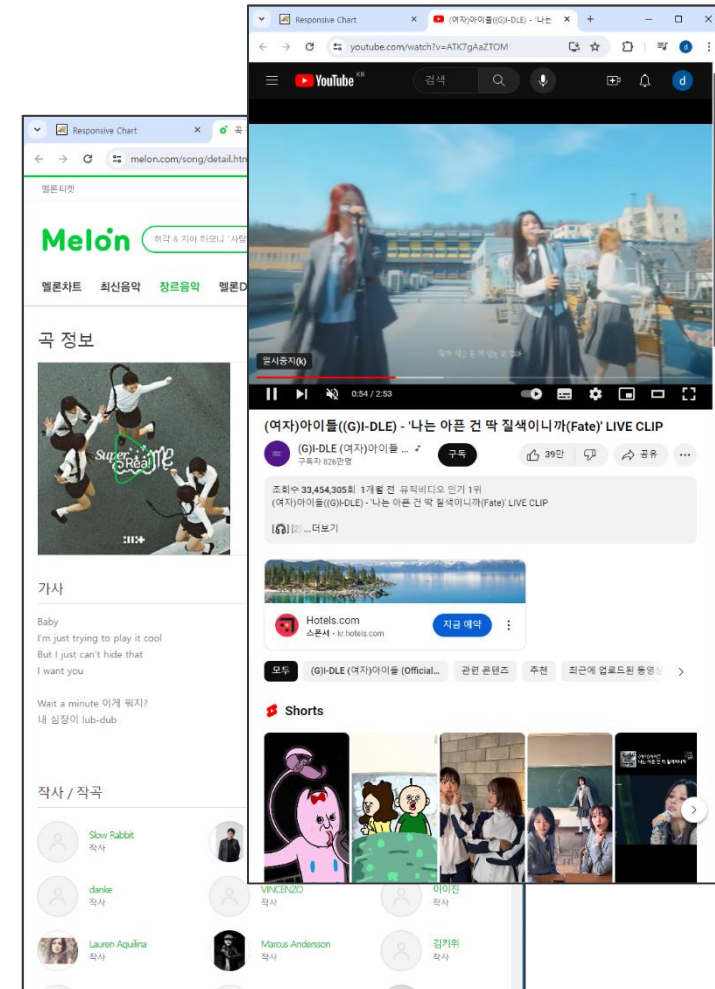
2. 플랫폼 별 음원 차트 서비스

2.8 해당 영역을 클릭하면 Melon과 Youtube의 경우 곡의 URL정보를 바탕으로 해당 플랫폼으로 이동이 가능합니다.



2024년 04월 29일 차트
Album 이미지를 클릭해보세요.

Rank	Album	Title	Artist	발행일
1	Magnetic	장르 : 랑스	이달릭(LULUT)	2024-03-25
2	(여자)아이들 - '나는 아픈 건 딱 질색이니까'	장르 : 랑스	(여자)아이들	2024-01-29
3	첫 만남은 계획대로 되지 않아	장르 : 랑스	TWS (투어스)	2024-01-22
4	그인종족	장르 : 록/에달	QWER	2024-04-01
5	천상연	장르 : 발라드	이달식	2024-02-21
6	빨간장	장르 : 힙합/소울	비비 (BIBI)	2024-02-13
7	Love suits all	장르 : 발라드	아이유	2024-01-24
8	미안해 미워해 사랑해	장르 : 발라드/곡내드라마	Crush	2024-03-24
9	엑셀러	장르 : 록/에달	DAY6 (데이식스)	2017-02-06
10	한 곡이자가 될 수 있게	장르 : 록/에달	DAY6 (데이식스)	2019-07-15

Melon

곡 정보

가사

작사 / 작곡

YouTube

(여자)아이들((G)-DLE) - '나는 아픈 건 딱 질색이니까(Fate)' LIVE CLIP

2. 플랫폼 별 음원 차트 서비스

2.9 멜론 크롤링

2.9.1 URL 확인 결과 시작일과 종료일로 구분하여 주 단위 데이터를 관리하고 있는 것을 확인

2.9.2 시작일과 종료일 설정을 통해 주count 계산

2.9.3 필요한 항목의 HTML을 확인하여 항목저장

The screenshot shows the Melon weekly chart page with a table of songs. The URL in the browser is `melon.com/chart/week/index.htm?classCd=GN0000&moved=Y&startDay=20240422&endDay=20240428`. The table lists songs like 'Magnetic' by SUPER REAL ME, '나는 아픈 건 막 질색이니까' by (여자)아이들, etc. The DevTools console shows the HTML structure, with red boxes highlighting the `tr` and `td` elements for the first row, and the `tbody` element for the entire table.

```
# 시작일과 종료일 설정
start_day = datetime(2023, 1, 2)
end_day = datetime(2024, 4, 28)

# 시작일과 종료일 사이의 주count 계산
total_weeks = ((end_day - start_day).days // 7) + 1

# 엑셀 파일 생성
wb = openpyxl.Workbook()
ws = wb.active
ws.append(["Day", "Rank", "Title", "Album_ing_urls", "album_code"])

# 주 단위로 날짜 출력
current_day = start_day
print(f"시작일 : {start_day}, 종료일 : {end_day}")
Week_Count = 0

while current_day <= end_day:
    # 주의 시작일과 끝날을 계산
    week_start = current_day.strftime("%Y%m%d")
    week_end = (current_day + timedelta(days=6)).strftime("%Y%m%d")
    current_day += timedelta(weeks=1)

    response = requests.get("https://www.melon.com/chart/week/index.htm?classCd=GN0000&moved=Y&startDay={0}&endDay={1}".format(week_start, week_end))

    URL = "https://www.melon.com/chart/week/index.htm?classCd=GN0000&moved=Y&startDay={0}&endDay={1}".format(week_start, week_end)

    print(URL)
    html = response.content.decode("utf-8", "replace")
    soup = BeautifulSoup(html, 'html.parser', from_encoding='cp949')

    my_st = soup.select(
        'tr>td>div>table>tbody'
    )

    for stock_name in my_st:
        # 앨범 코드 출력
        Album_codes = []
        for tag in soup.find_all('tr'):
            if 'class' in tag.attrs:
                if 'lst50' in tag['class'] or 'lst100' in tag['class']:
                    Album_codes.append(tag['data-song-no'])

        # 순위
        Rank = stock_name.select("td:nth-child(2) > div > span.rank")
        Rank_Num = [int(Rank.text) for Rank in Rank]

        # 제목
        Title = stock_name.select("td:nth-child(4) > div > div.ellipsis.rank01 > span > a")
        Title_Text = [Title.text for Title in Title]

        # 앨범 이미지
        Album_ing_urls = [img['src'] for img in stock_name.select("td:nth-child(4) > div > a > img")]

        for rank, title_Text, album_ing_urls, album_code in zip(Rank_Num, Title_Text, Album_ing_urls, Album_codes):
            # print(week_start, rank, name_text, prod_text, album_ing_urls, only_codes)
            # print("-----")
            ws.append([week_start, rank, title_Text, album_ing_urls, album_code])

    # 주별 카운트 확인
    Week_Count += 1
    # 주별당 count : 총Count
    print(f"Day_Count : {Week_Count}, Total_Count : {total_weeks}")

# 엑셀 파일 저장
file_path = f"C:\P_Project\3-Project\3-Project_data\3-Melon_Crawling_No_Detail_4_30.xlsx"
wb.save(file_path)
print(f"엑셀 파일이 저장되었습니다.", file_path)
```

시작일 : 2023-01-02 00:00:00, 종료일 : 2024-04-28 00:00:00
<https://www.melon.com/chart/week/index.htm?classCd=GN0000&moved=Y&startDay=20230102&endDay=20230108>
 Day_Count : 1, Total_Count : 70
<https://www.melon.com/chart/week/index.htm?classCd=GN0000&moved=Y&startDay=20230109&endDay=20230115>
 Day_Count : 2, Total_Count : 70
<https://www.melon.com/chart/week/index.htm?classCd=GN0000&moved=Y&startDay=20230116&endDay=20230122>
 Day_Count : 3, Total_Count : 70
<https://www.melon.com/chart/week/index.htm?classCd=GN0000&moved=Y&startDay=20230123&endDay=20230129>
 Day_Count : 4, Total_Count : 70
<https://www.melon.com/chart/week/index.htm?classCd=GN0000&moved=Y&startDay=20230130&endDay=20230205>
 Day_Count : 5, Total_Count : 70
<https://www.melon.com/chart/week/index.htm?classCd=GN0000&moved=Y&startDay=20230206&endDay=20230212>
 Day_Count : 6, Total_Count : 70
<https://www.melon.com/chart/week/index.htm?classCd=GN0000&moved=Y&startDay=20230213&endDay=20230219>
 Day_Count : 7, Total_Count : 70
<https://www.melon.com/chart/week/index.htm?classCd=GN0000&moved=Y&startDay=20230220&endDay=20230226>
 Day_Count : 8, Total_Count : 70
<https://www.melon.com/chart/week/index.htm?classCd=GN0000&moved=Y&startDay=20230227&endDay=20230305>
 Day_Count : 9, Total_Count : 70
<https://www.melon.com/chart/week/index.htm?classCd=GN0000&moved=Y&startDay=20230306&endDay=20230312>
 Day_Count : 10, Total_Count : 70
<https://www.melon.com/chart/week/index.htm?classCd=GN0000&moved=Y&startDay=20230313&endDay=20230319>
 Day_Count : 11, Total_Count : 70
<https://www.melon.com/chart/week/index.htm?classCd=GN0000&moved=Y&startDay=20230320&endDay=20230326>
 Day_Count : 12, Total_Count : 70
 ...
 Day_Count : 60, Total_Count : 70
<https://www.melon.com/chart/week/index.htm?classCd=GN0000&moved=Y&startDay=20240429&endDay=20240405>
 Day_Count : 70, Total_Count : 70
 Excel 파일이 저장되었습니다.: C:\P_Project\3-Project\3-Project_data\3-Melon_Crawling_No_Detail_4_30.xlsx

2. 플랫폼 별 음원 차트 서비스

2.9 멜론 크롤링

2.9.4 앨범 코드를 활용하여 상세페이지에서 장르 발매일 등 상세정보 크롤링 가능

※ 2.1.1에서 앨범코드를 크롤링하여 해당 상세페이지 정보 크롤링 가능

곡 정보

Magnetic
아일릿(ILLIT)

앨범 SUPER REAL ME
발매일 2024.03.25
장르 댄스
FLAC Flac 16/24bit

93,547
곡 다운 > FLAC 다운 > 선물하기 >

가사

```
for detail in Unique_Album_codes:
    response_Detail = requests.get("https://www.melon.com/song/detail.htm?songId={}".format(detail), headers=headers)
    URL_Detail = "https://www.melon.com/song/detail.htm?songId={}".format(detail)

    html_Detail = response_Detail.content.decode('utf-8', 'replace')
    soup_Detail = BeautifulSoup(html_Detail, 'html.parser', from_encoding='cp949')

    my_st_Detail = soup_Detail.select(
        '#downloadfrm > div > div > div.entry'
    )
    for stock_name in my_st_Detail:
        # 발매일
        Release = stock_name.select("#downloadfrm > div > div > div.entry > div.meta > dl > dd:nth-child(4)")
        Release_date = [Release.text for Release in Release]

        # 장르
        Genre = stock_name.select("#downloadfrm > div > div > div.entry > div.meta > dl > dd:nth-child(6)")
        Genre_Text = [Genre.text for Genre in Genre]

        # 가수
        Singer_name = stock_name.select("#downloadfrm > div > div > div.entry > div.info > div.artist > a > span:nth-child(1)")
        Singer = [Singer_name.text for Singer_name in Singer_name]

        # 데이터 추가
        for detail_code, singer, release_date, genre_text in zip(Unique_Album_codes, Singer, Release_date, Genre_Text):
            ws.append([detail, singer, release_date, genre_text])

    print(detail, Singer, Release_date, Genre_Text)

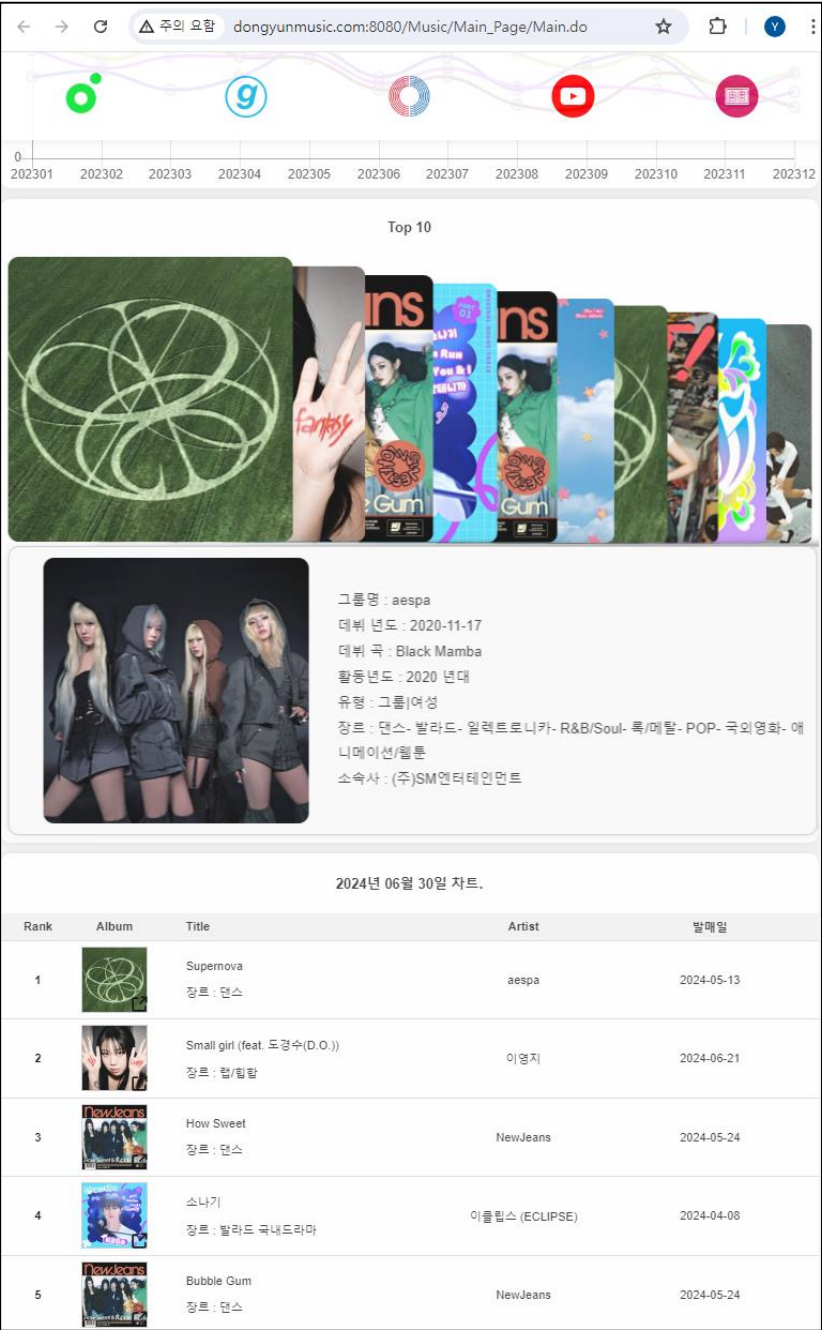
# 엑셀 파일 저장
file_path = r"C:\P_Project\1.Project\5-Project_data\1.Melon_Crawling_Detail.4.29.xlsx"
wb.save(file_path)
print("Excel 파일이 저장되었습니다.", file_path)

34061322 ['임영웅'] ['2021.10.11'] ['발라드, 국내드라마']
33496587 ['경서예지', '전건호'] ['2021.05.10'] ['발라드']
35008524 ['임영웅'] ['2022.05.02'] ['발라드']
35008525 ['임영웅'] ['2022.05.02'] ['록/메탈']
35008527 ['임영웅'] ['2022.05.02'] ['발라드']
32872978 ['방탄소년단'] ['2020.08.24'] ['댄스']
32061975 ['AKMU (악뮤)'] ['2019.09.25'] ['발라드']
37222942 ['아이유'] ['2024.02.20'] ['랩/힙합']
37138464 ['(여자)아이들'] ['2024.01.29'] ['댄스']
36855841 ['임영웅'] ['2023.10.09'] ['댄스']
37138469 ['(여자)아이들'] ['2024.01.29'] ['댄스']
37138471 ['(여자)아이들'] ['2024.01.22'] ['댄스']
34657844 ['멜로망스'] ['2022.02.18'] ['발라드, 국내드라마']
37064763 ['태연 (TAEYEON)'] ['2023.12.17'] ['발라드, 국내드라마']
37229121 ['아이유'] ['2024.02.20'] ['록/메탈']
30244931 ['방탄소년단'] ['2017.02.13'] ['랩/힙합']
34908740 ['임영웅'] ['2022.05.02'] ['발라드']
37431365 ['RIIZE'] ['2024.04.18'] ['댄스']
37347911 ['아일릿(ILLIT)'] ['2024.03.25'] ['댄스']
37426262 ['Crush'] ['2024.04.17'] ['R&B/Soul']
34431086 ['김민석'] ['2021.12.19'] ['발라드']
36617841 ['정국'] ['2023.07.14'] ['댄스']
36416114 ['aespa'] ['2023.05.08'] ['댄스']
36877939 ['QWER'] ['2023.10.18'] ['록/메탈']
37053556 ['이무진'] ['2023.12.13'] ['록/메탈']
...
36936167 ['정국'] ['2023.11.03'] ['댄스']
31927275 ['DAV6 (데이식스)'] ['2019.07.15'] ['록/메탈']
36871671 ['IVE (아이브)'] ['2023.10.13'] ['댄스']
Excel 파일이 저장되었습니다: C:\P_Project\1.Project\5-Project_data\1.Melon_Crawling_Detail.4.29.xlsx
```

2. 플랫폼 별 음원 차트 서비스

2.10 Top10 상세정보

2.10.1 아티스트 코드를 활용하여 상세정보 크롤링



2. 플랫폼 별 음원 차트 서비스

2.10 Top10 상세정보

2.10.2 Mysql과 연동하여 상위10개의 데이터만 추출히하여 Singer_Codes 배열에 삽입

```
import mysql.connector

# 데이터베이스 연결
conn = mysql.connector.connect(
    host='localhost',
    user='root',
    password='music',
    database='MusicDatabase'
)

cursor = conn.cursor()

# 쿼리 실행 기준에 존재하는 데이터는 제외하고 출력
query = """
    SELECT Singer_Code
    FROM Music
    WHERE Day = (SELECT max(Day) FROM Music)
    AND Ranking < 11
    AND Singer_Code not in (select Group_Code from idol_Group group by Group_Code)
    AND Singer_Code IS NOT NULL
    GROUP BY Singer_Code
    ORDER BY MIN(Ranking) ASC;
"""
cursor.execute(query)

# 결과 출력
Singer_Code_List = cursor.fetchall()
for (Singer_Code) in Singer_Code_List:
    print(Singer_Code)

# 커서와 연결 닫기
cursor.close()
conn.close()
```

[2] ✓ 0.2s Python

```
...
(2899555,)
(2636153,)
(3114174,)
(3752136,)
(3618529,)
(602056,)
(3055146,)
(3743525,)

```

```
Singer_Code_List

# 튜플의 첫 번째 요소만 추출
singer_codes = [code[0] for code in Singer_Code_List]

print(singer_codes)
```

[3] ✓ 0.0s Python

```
...
[2899555, 2636153, 3114174, 3752136, 3618529, 602056, 3055146, 3743525]
```


2. 플랫폼 별 음원 차트 서비스

2.10 Top10 상세정보

2.10.3 멜론 웹사이트에서 아티스트 정보를 활용하여 데이터를 추출, 이를 엑셀 파일로 저장하는 작업을 수행합니다

아티스트 채널

aespa
카리나 (KARINA), 자엘 (GISELLE), 윈터 (WINTER), 닝닝 (NINGNING)

데뷔: 2020.11.17 | Black Mamba
활동유형: 그룹
소속사: (주)SM엔터테인먼트
수상이력: 2024 대한민국 퍼스트브랜드 대상 | 여자아이돌

★ **관망기** 134,939명

세오이, Adon, NING, GISELLE

나만의 채널도 더보기

수상이력

2024 대한민국 퍼스트브랜드 대상 | 여자아이돌
2023 한류유치 어워드 | 글로벌 제너레이션 아이콘상
2023 한류유치 어워드 | 베스트 브랜드 리더상

아티스트 소개

aespa는 SM 엔터테인먼트 소속의 걸그룹으로 카리나 (KARINA), 자엘 (GISELLE), 윈터 (WINTER), 닝닝 (NINGNING)으로 구성되어 있다. 팀명 'aespa'는 'Avatar X Experience'를 표현한 'a'와 '와'의 합성어라는 뜻의 합성어 'aspect'를 결합해 만든 이름으로, '자신'의 또 다른 자아인 아바타를 만나 새로운 세계를 경험하게 된다는 세계관을 바탕으로 특징적이고 다채로운 활동을 선보일 예정이다.

활동정보

대위	활동년대	유형	장르	소속사명
대위곡	2020.11.17	2020년대	그룹 여성	대위곡
장르	댄스, 발라드, 일렉트로니카, R&B/Soul, 록/메탈, POP, 국악영화, 애니메이션/웹툰			
소속사명	이선/웹툰			
	(주)SM엔터테인먼트			

그룹멤버

카리나 (KARINA) | 닝닝 (NINGNING) | 자엘 (GISELLE) | 윈터 (WINTER)

대한민국/여성/솔로 | 발라드, 댄스, R&B/S... | 대한민국/여성/솔로 | 발라드, 댄스, R&B/S... | 대한민국/여성/솔로 | 발라드, 댄스, R&B/S... | 대한민국/여성/솔로 | 발라드, 댄스, R&B/S...

★ **관망기** 10,280 | ★ **관망기** 5,508 | ★ **관망기** 15,652 | ★ **관망기** 6,173

```
# 엑셀 파일 생성
wb = openpyxl.Workbook()
ws = wb.active
ws.append(["Group_Code", "Group_Name", "Group_Ingurl", "debut_song", "debut_year", "Year_Active", "Group_type", "Genre_Text", "Company"])

# User-Agent 헤더
headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.3'
}

for i in singer_codes:
    response = requests.get("https://www.melon.com/artist/detail.htm?artistid={0}".format(i), headers=headers)

    url = "https://www.melon.com/artist/detail.htm?artistid={0}".format(i)
    print(url)
    html = response.content.decode('utf-8', 'replace')
    soup = BeautifulSoup(html, 'html.parser', from_encoding='cp949')

    my_st = soup.select("#conts")

    # print(i) 그룹코드

    for stock_name in my_st:
        img_tag = soup.select_one("#artistimgarea img")
        img_url = img_tag["src"] if img_tag else ""

        print("-----")

        # 그룹명
        artist_tag = soup.select_one("#title_artist") # 그룹소명이 title_artist인 요소 선택
        Group_Name = artist_tag.find("strong").next_sibling.strip() # strong 태그 다음의 텍스트 가져오기

        # 데뷔 연도 가져오기
        debut_year_tag = soup.find("dt", text="데뷔")
        # null 처리
        debut_year_text = debut_year_tag.find_next_sibling("dd").get_text(strip=True) if debut_year_tag else None

        if debut_year_text: # debut_year_text가 None이 아닌 경우에만 실행
            # '국채상' 기준으로 문자열 분리
            parts = debut_year_text.split("국채상")

            # 신인가수의 경우 데뷔곡이 없어 예외 발생
            if len(parts) >= 2:
                # 데뷔 연도 추출
                debut_year = parts[0].strip() # 공백 제거

                # 데뷔 곡 추출
                debut_song = parts[1].strip() # 공백 제거

        year_active_tag = soup.find("dt", text="활동년대")
        year_active = year_active_tag.find_next_sibling("dd").get_text(strip=True) if year_active_tag else None

        group_type_tag = soup.find("dt", text="유형")
        group_type = group_type_tag.find_next_sibling("dd").get_text(strip=True) if group_type_tag else None

        genre_tag = soup.find("dt", text="장르")
        genre_text = genre_tag.find_next_sibling("dd").get_text(strip=True) if genre_tag else None

        company_tag = soup.find("dt", text="소속사명")
        company = company_tag.find_next_sibling("dd").get_text(strip=True) if company_tag else None

        # 출력
        print(f"Group_Code = {i}")
        print(f"Group_Name = {Group_Name}")
        print(f"Group_Ingurl = {img_url}")
        print(f"debut_year = {debut_year}")
        print(f"debut_song = {debut_song}")
        print(f"Year_Active = {year_active}")
        print(f"Group_type = {group_type}")
        print(f"Genre_Text = {genre_text}")
        print(f"Company = {company}")

        ws.append([i, Group_Name, img_url, debut_song, debut_year, year_active, group_type, genre_text, company])

# 엑셀 파일 저장
file_path = r"C:\P_Project\11-Project\5-Project_data\7.1dol_Group.xlsx"
wb.save(file_path)
print(f"엑셀 파일이 저장되었습니다." file_path)
```

AWS EC2 배포

2. 플랫폼 별 음원 차트 서비스

3.1 AWS 서버 연결

3.1.1 ubuntu를 선택한 이유는 오픈소스 운영체제로 라이선스 비용이 들지 않아 예산이 절약되며,
ubuntu는 가장 인기있는 운영체제로 문제 발생 시 해결할 수 있는 자료를 쉽게 찾을 수 있습니다.

3.1.2 키페어 생성은 인스턴스에 안전하게 연결할 수 있도록 하는 보안 메커니즘입니다

The screenshot shows the AWS Management Console interface for launching an EC2 instance. The 'Launch an instance' page is displayed, with the 'Ubuntu' AMI selected in the 'Quick Start' section. The 'Key Pair' section shows a key pair named 'music' is selected. A blue notification box on the right states that the t2.micro instance type is not available in the selected region.

요약

- 인스턴스 개수: 1
- 소프트웨어 이미지(AMI): Canonical, Ubuntu, 24.04 LTS, ... 더 보기
ami-0e6f2b2fa0ca704d0
- 가상 서버 유형(인스턴스 유형): t2.micro
- 방화벽(보안 그룹): 새 보안 그룹
- 스토리지(볼륨): 1개의 볼륨 - 8GiB

프리 티어: 첫 해에는 월별 프리 티어
AMI에 대한 t2.micro(또는 t2.micro)를 사용할 수 없는 리전의 t3.micro) 인스턴스 사용량 750시간, 월별 퍼블릭 IPv4 주소 사용량 750시간, EBS 스토리지 30GiB, IO 2백만 개, 스냅샷 1GB, 인터넷 대역폭 100GB가 포함됩니다.

키 페어(로그인) 정보

키 페어를 사용하여 인스턴스에 안전하게 연결할 수 있습니다. 인스턴스를 시작하기 전에 선택한 키 페어에 대한 액세스 권한이 있는지 확인하세요.

키 페어 이름 - 필수
music

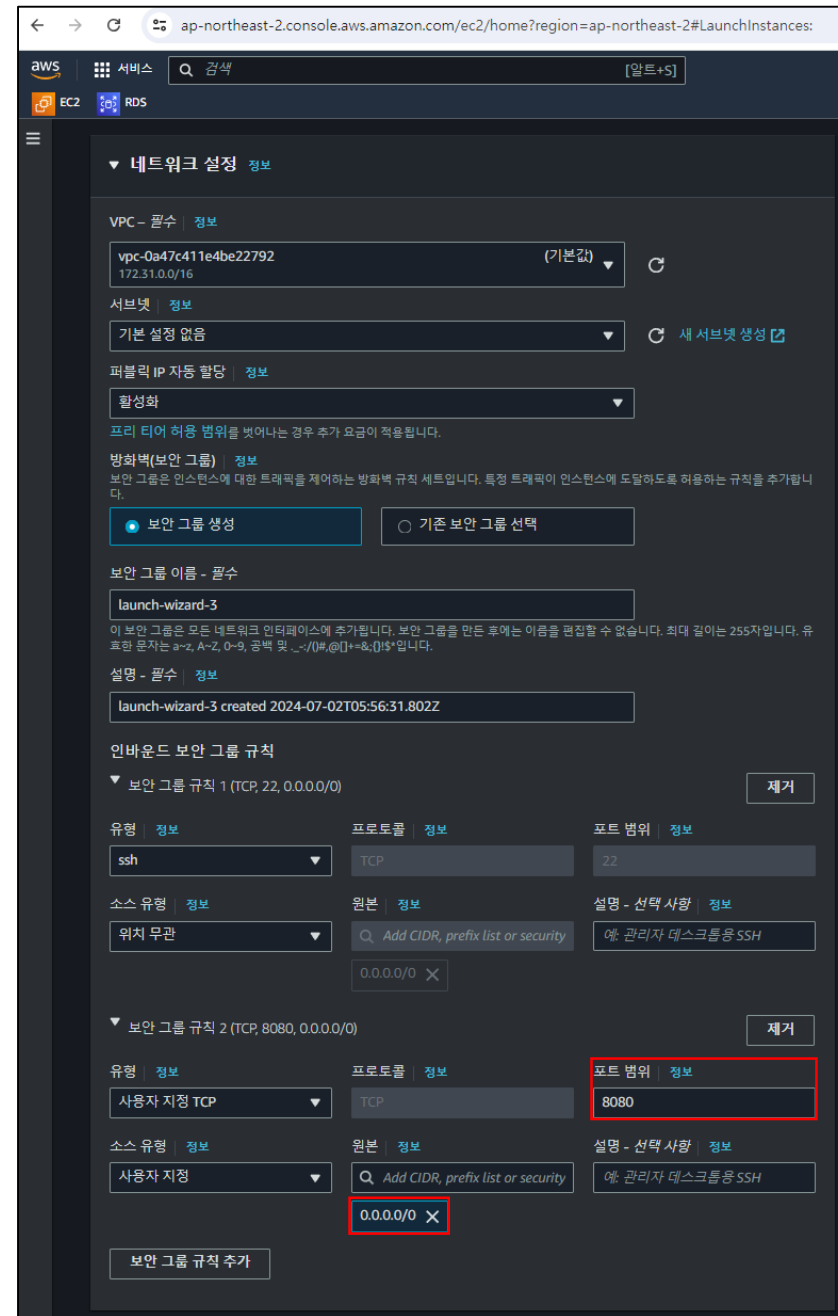
새 키 페어 생성

2. 플랫폼 별 음원 차트 서비스

3.1 AWS 서버 연결

3.1.3 톰캣서버의 포트는 8080 입니다. 따라서 포트 범위를 8080으로 설정하였으며, 외부에서 해당 포트를 통해 웹 사이트를 접근할 수 있습니다.

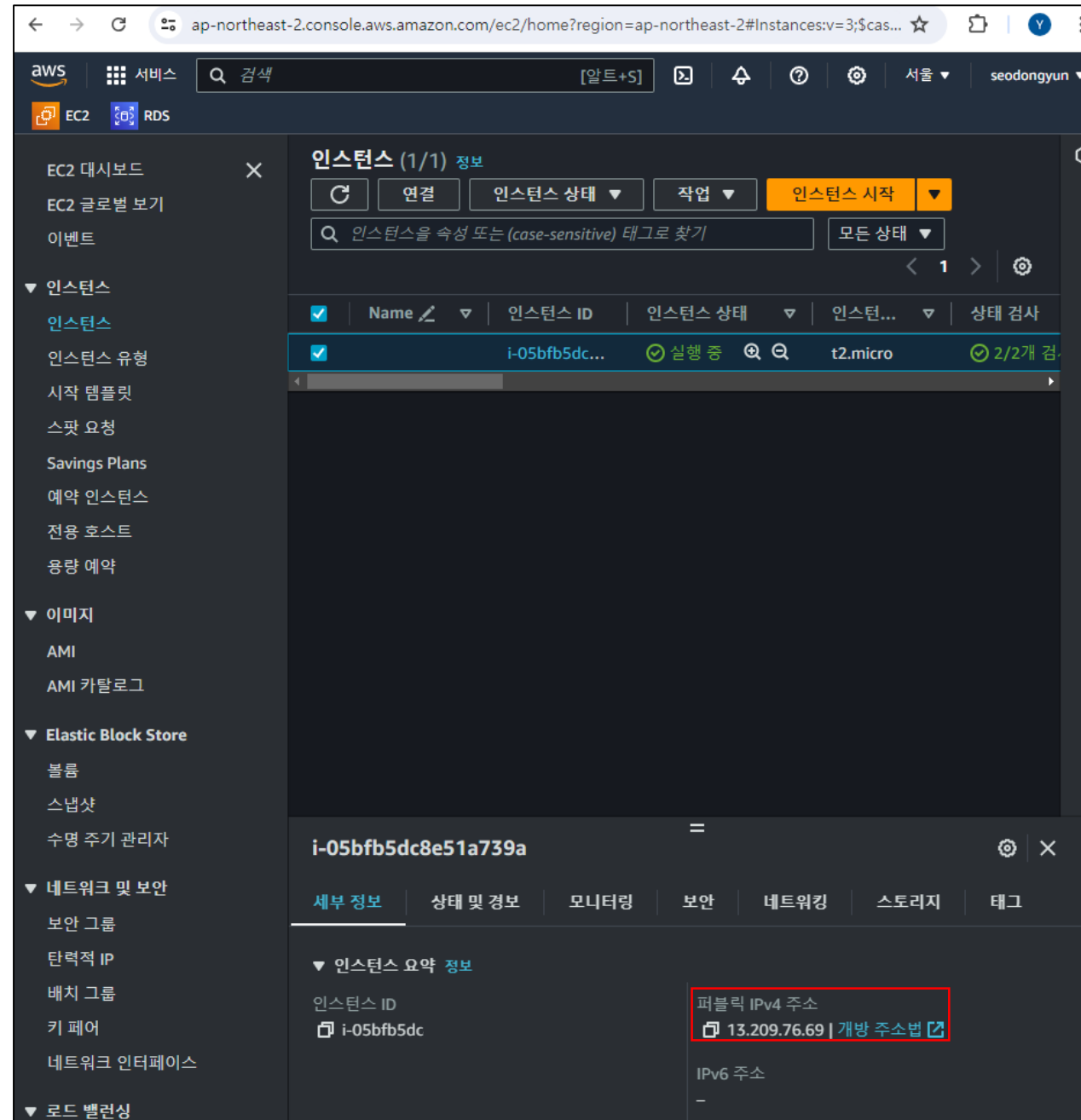
3.1.4 IP 주소로부터의 접근을 허용하여 해당 포트로 접근이 가능하게 됩니다.



2. 플랫폼 별 음원 차트 서비스

3.1 AWS 서버 연결

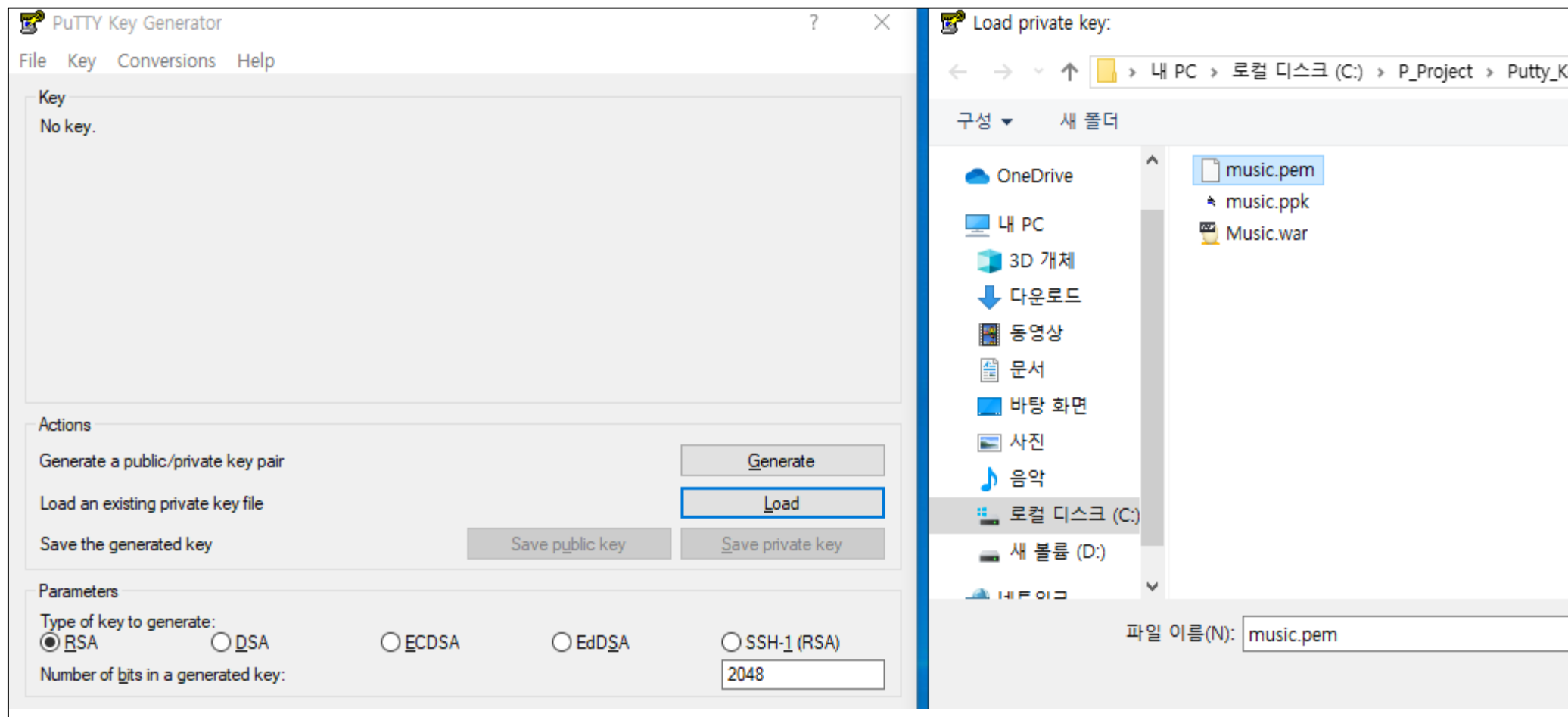
3.1.5 성공적으로 인스턴스를 생성하였으며 퍼블릭 IPv4 주소를 통해 서버에 접속할 수 있습니다.



2. 플랫폼 별 음원 차트 서비스

3.1 AWS 서버 연결

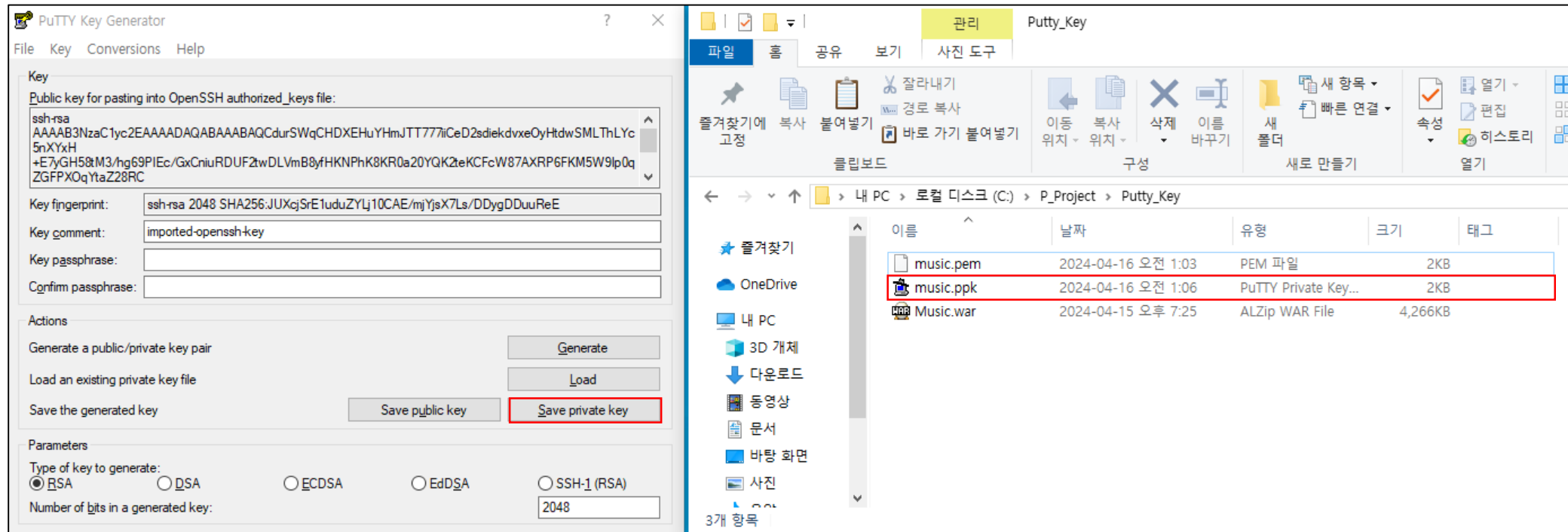
3.1.6 Putty Key Gen을 실행하여 3.1.2에서 생성한 키 페어를 선택합니다.



2. 플랫폼 별 음원 차트 서비스

3.1 AWS 서버 연결

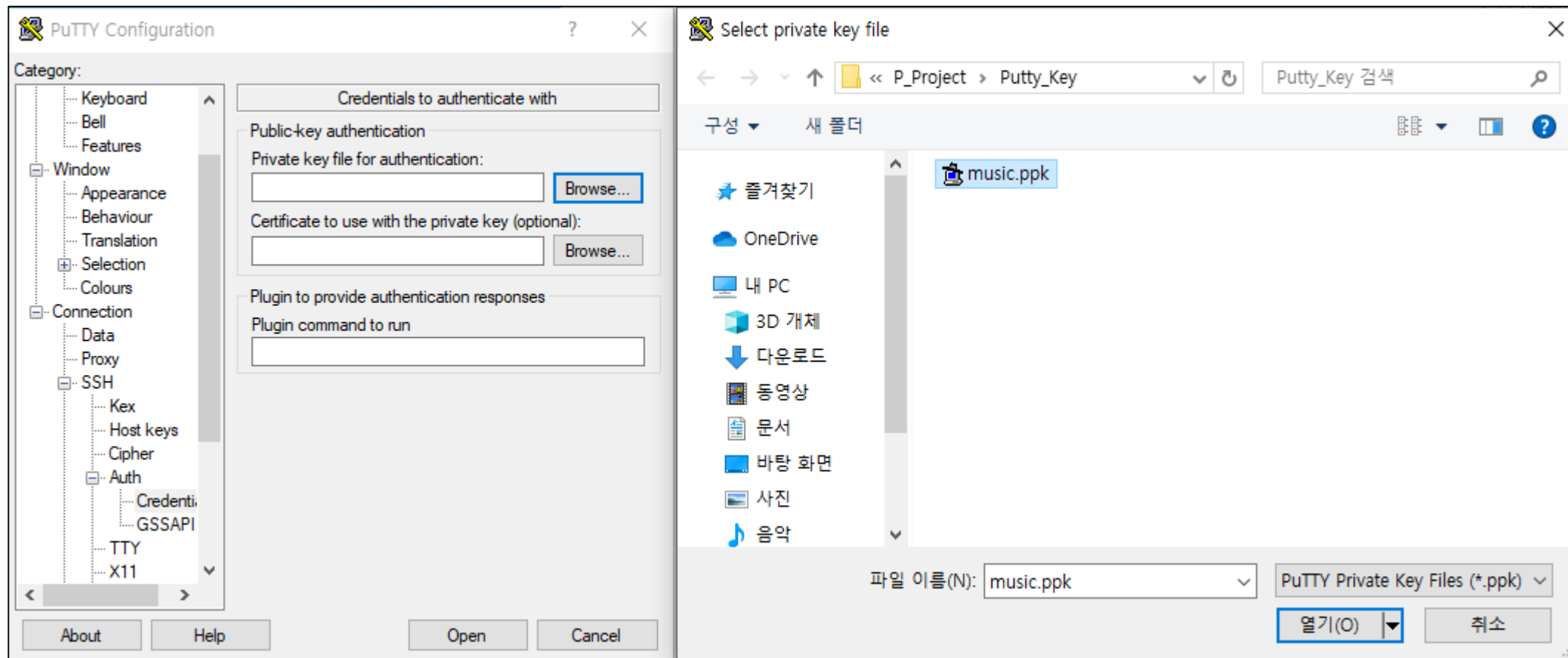
3.1.7 Save private key를 저장하면 ppk 파일이 생성됩니다.



2. 플랫폼 별 음원 차트 서비스

3.1 AWS 서버 연결

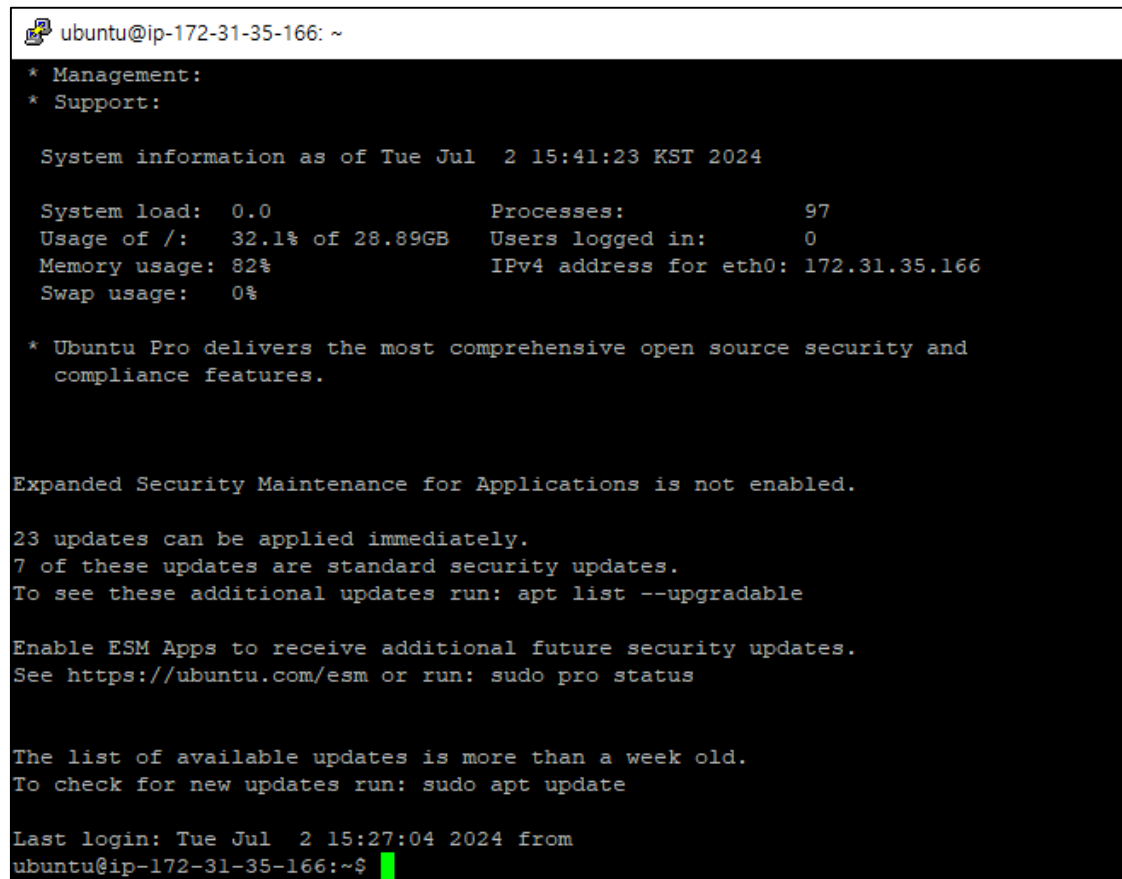
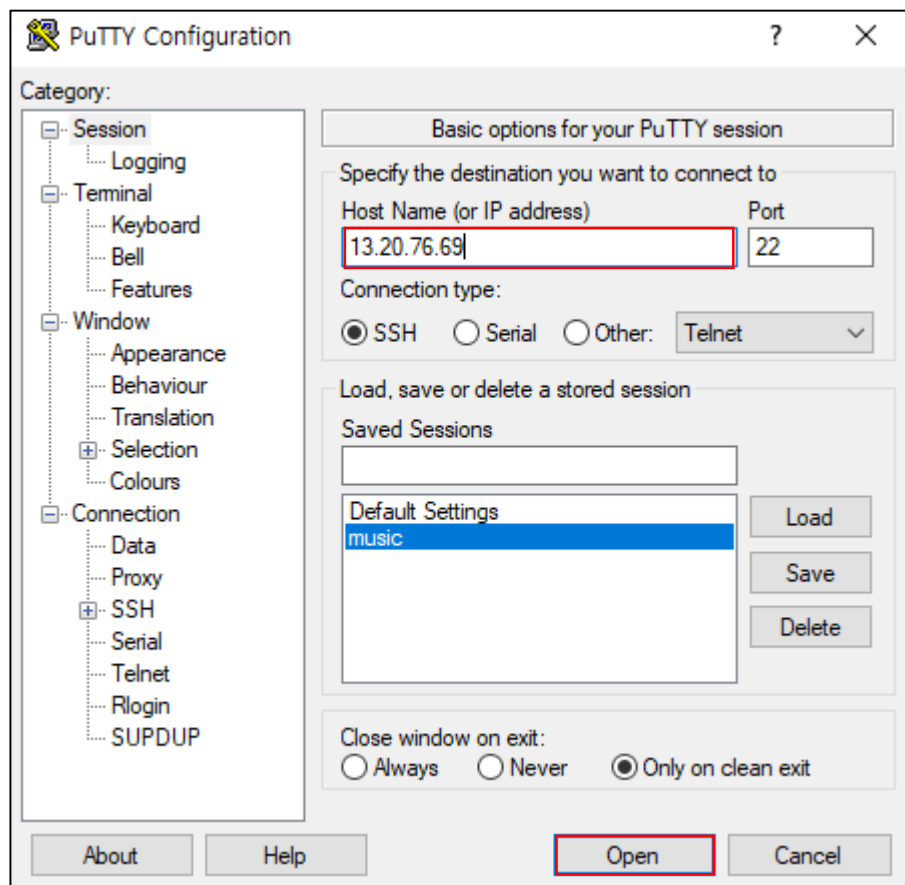
3.1.8 Connection->SSH->Auth->Credentials를 눌러 Private key file for authentication에서 Browse를 눌러 아까 생성한 ppk파일을 지정합니다.



2. 플랫폼 별 음원 차트 서비스

3.1 AWS 서버 연결

3.1.9 Host Name 에 IPv4 주소를 입력하여 Open을 클릭합니다.



2. 플랫폼 별 음원 차트 서비스

3.1 AWS 서버 연결

3.1.10 Ubuntu JDK설치하기 전 `sudo vi /etc/apt/sources.list` 파일을

`:%s/ap-northeast-2.ec2.archive.ubuntu.com/mirror.kakao.com` 해당 명령어로 수정합니다.

```
ubuntu@ip-172-31-35-166: ~  
## team. Also, please note that software in universe WILL NOT receive any  
## review or updates from the Ubuntu security team.  
deb http://mirror.kakao.com/ubuntu/ jammy universe  
# deb-src http://mirror.kakao.com/ubuntu/ jammy universe  
deb http://mirror.kakao.com/ubuntu/ jammy-updates universe  
# deb-src http://mirror.kakao.com/ubuntu/ jammy-updates universe  
  
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu  
## team, and may not be under a free licence. Please satisfy yourself as to  
## your rights to use the software. Also, please note that software in  
## multiverse WILL NOT receive any review or updates from the Ubuntu  
## security team.  
deb http://mirror.kakao.com/ubuntu/ jammy multiverse  
# deb-src http://mirror.kakao.com/ubuntu/ jammy multiverse  
deb http://mirror.kakao.com/ubuntu/ jammy-updates multiverse  
# deb-src http://mirror.kakao.com/ubuntu/ jammy-updates multiverse  
  
## N.B. software from this repository may not have been tested as  
## extensively as that contained in the main release, although it includes  
## newer versions of some applications which may provide useful features.  
## Also, please note that software in backports WILL NOT receive any review  
## or updates from the Ubuntu security team.  
deb http://mirror.kakao.com/ubuntu/ jammy-backports main restricted universe multiverse  
# deb-src http://mirror.kakao.com/ubuntu/ jammy-backports main restricted universe multiverse  
  
deb http://security.ubuntu.com/ubuntu jammy-security main restricted  
# deb-src http://security.ubuntu.com/ubuntu jammy-security main restricted  
deb http://security.ubuntu.com/ubuntu jammy-security universe  
# deb-src http://security.ubuntu.com/ubuntu jammy-security universe  
"/etc/apt/sources.list" 50L, 2766B
```

※ mirror.kakao.com과 같은 로컬 미러 서버를 사용하면 패키지 다운로드 속도가 크게 향상되며 카카오에서 운영하는 미러 서버로, 안정적이고 신뢰할 수 있어 수정이 필요합니다.

2. 플랫폼 별 음원 차트 서비스

3.1 AWS 서버 연결

3.1.11 apt-cache search jdk 해당 명령어를 통해 JDK리스트가 있는지 확인 후 설치

```
substance-flamingo-doc - Substance Flamingo plugin (documentation)
testng - testing framework for Java
uwsgi-app-integration-plugins - plugins for integration of uWSGI and application
uwsgi-plugin-jvm-openjdk-11 - Java plugin for uWSGI (OpenJDK 11)
uwsgi-plugin-jwsgi-openjdk-11 - JWSGI plugin for uWSGI (OpenJDK 11)
uwsgi-plugin-ring-openjdk-11 - Closure/Ring plugin for uWSGI (OpenJDK 11)
uwsgi-plugin-servlet-openjdk-11 - JWSGI plugin for uWSGI (OpenJDK 11)
visualvm - All-in-One Java Troubleshooting Tool
java-package - Utility for creating Java Debian packages
libasmtools-java - OpenJDK AsmTools
libtestng7-java - testing framework for Java
openjdk-19-dbg - Java runtime based on OpenJDK (debugging symbols)
openjdk-19-demo - Java runtime based on OpenJDK (demos and examples)
openjdk-19-doc - OpenJDK Development Kit (JDK) documentation
openjdk-19-jdk - OpenJDK Development Kit (JDK)
openjdk-19-jdk-headless - OpenJDK Development Kit (JDK) (headless)
openjdk-19-jre - OpenJDK Java runtime, using Hotspot JIT
openjdk-19-jre-headless - OpenJDK Java runtime, using Hotspot JIT (headless)
openjdk-19-jre-zero - Alternative JVM for OpenJDK, using Zero
openjdk-19-source - OpenJDK Development Kit (JDK) source files
openjdk-8-jdk - OpenJDK Development Kit (JDK)
openjdk-8-jdk-headless - OpenJDK Development Kit (JDK) (headless)
openjdk-8-jre-zero - Alternative JVM for OpenJDK, using Zero
ubuntu@ip-172-31-36-18:~$
```

2. 플랫폼 별 음원 차트 서비스

3.1 AWS 서버 연결

3.1.12 sudo apt-get install openjdk-19-jdk 명령어를 통해 jdk 설치

```
libdeflate0 libdrm-amdgpu libdrm-intel1 libdrm-nouveau2 libdrm-radeon1
libfontconfig1 libfontenc1 libgail-common libgail18 libgdk-pixbuf-2.0-0
libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common libgif7 libgll libgll-amber-dri
libgll-mesa-dri libglapi-mesa libglvnd0 libglx-mesa0 libglx0 libgraphite2-3
libgtk2.0-0 libgtk2.0-bin libgtk2.0-common libharfbuzz0b libice-dev libice6
libjbig0 libjpeg-turbo8 libjpeg8 liblcms2-2 libllvml5 libpango-1.0-0
libpangocairo-1.0-0 libpangoft2-1.0-0 libpciaccess0 libpcsclite1
libpixmap-1-0 libpthread-stubs0-dev librsvg2-2 librsvg2-common
libsensors-config libsensors5 libsm-dev libsm6 libthai-data libthai0
libtiff5 libwebp7 libx11-dev libx11-xcb1 libxau-dev libxaw7 libxcb-dri2-0
libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-render0
libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxcbl-dev
libxcompositel libxcursor1 libxdamage1 libxdmcp-dev libxfixes3 libxft2
libxi6 libxinerama1 libxkbfile1 libxmu6 libxpm4 libxrandr2 libxrender1
libxshmfence1 libxt-dev libxt6 libxtst6 libxv1 libxxf86dgal libxxf86vml
openjdk-19-jdk openjdk-19-jdk-headless openjdk-19-jre
openjdk-19-jre-headless session-migration ubuntu-mono x11-common x11-utils
x11proto-dev xorg-sgml-doctools xtrans-dev
The following packages will be upgraded:
  libx11-6
1 upgraded, 127 newly installed, 0 to remove and 128 not upgraded.
Need to get 369 MB of archives.
After this operation, 724 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

2. 플랫폼 별 음원 차트 서비스

3.1 AWS 서버 연결

3.1.13 JDK와 동일하게 패키지 리스트에 Tomcat 존재 여부 확인

```
libnetty-tcnative-java - Tomcat native fork for Netty
libnetty-tcnative-jni - Tomcat native fork for Netty (JNI library)
libspring-instrument-java - modular Java/J2EE application framework - Instrumentation
libtcnative-1 - Tomcat native library using the Apache Portable Runtime
libtomcat9-embed-java - Apache Tomcat 9 - Servlet and JSP engine -- embed libraries
libtomcat9-java - Apache Tomcat 9 - Servlet and JSP engine -- core libraries
libtomcatjss-java - JSSE implementation using JSS for Tomcat
monitoring-plugins-contrib - Plugins for nagios compatible monitoring systems
python3-ajpy - Python module to craft AJP requests
resource-agents-extra - Cluster Resource Agents
tomcat-jakartaee-migration - Apache Tomcat migration tool for Jakarta EE
tomcat9 - Apache Tomcat 9 - Servlet and JSP engine
tomcat9-admin - Apache Tomcat 9 - Servlet and JSP engine -- admin web applications
tomcat9-common - Apache Tomcat 9 - Servlet and JSP engine -- common files
tomcat9-docs - Apache Tomcat 9 - Servlet and JSP engine -- documentation
tomcat9-examples - Apache Tomcat 9 - Servlet and JSP engine -- example web applications
tomcat9-user - Apache Tomcat 9 - Servlet and JSP engine -- tools to create user instances
yasat - simple stupid audit tool
ubuntu@ip-172-31-35-166:~$
```

2. 플랫폼 별 음원 차트 서비스

3.1 AWS 서버 연결

3.1.14 Tomcat 10 이상의 버전이 존재하지않아 외부 링크를 통해 다운로드 진행

10.1.13

Please see the [README](#) file for packaging information. It

Binary Distributions

- Core:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
 - [32](#) 새 탭에서 링크 열기
 - [64](#) 새 창에서 링크 열기
 - [32](#) 시크릿 창에서 링크 열기
- Full doc
 - [tar](#) ☐ 듀얼 탭에서 링크 열기
- Deploye
 - [zip](#) ☐ 모바일 창에서 링크 열기
 - [tar](#) ☐ 사이드바에서 링크 열기
- Embedd
 - [tar](#) 다른 이름으로 링크 저장...
 - [zip](#) 링크 주소 복사

Source Code Di

검사

```
ubuntu@ip-172-31-35-166:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-35-166:~$ curl --output tomcat.tar.gz https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.13/bin/apache-tomcat-10.1.13.tar.gz
```

```
ubuntu@ip-172-31-35-166:~$ ls
tomcat.tar.gz
ubuntu@ip-172-31-35-166:~$ tar -xvf tomcat.tar.gz
```

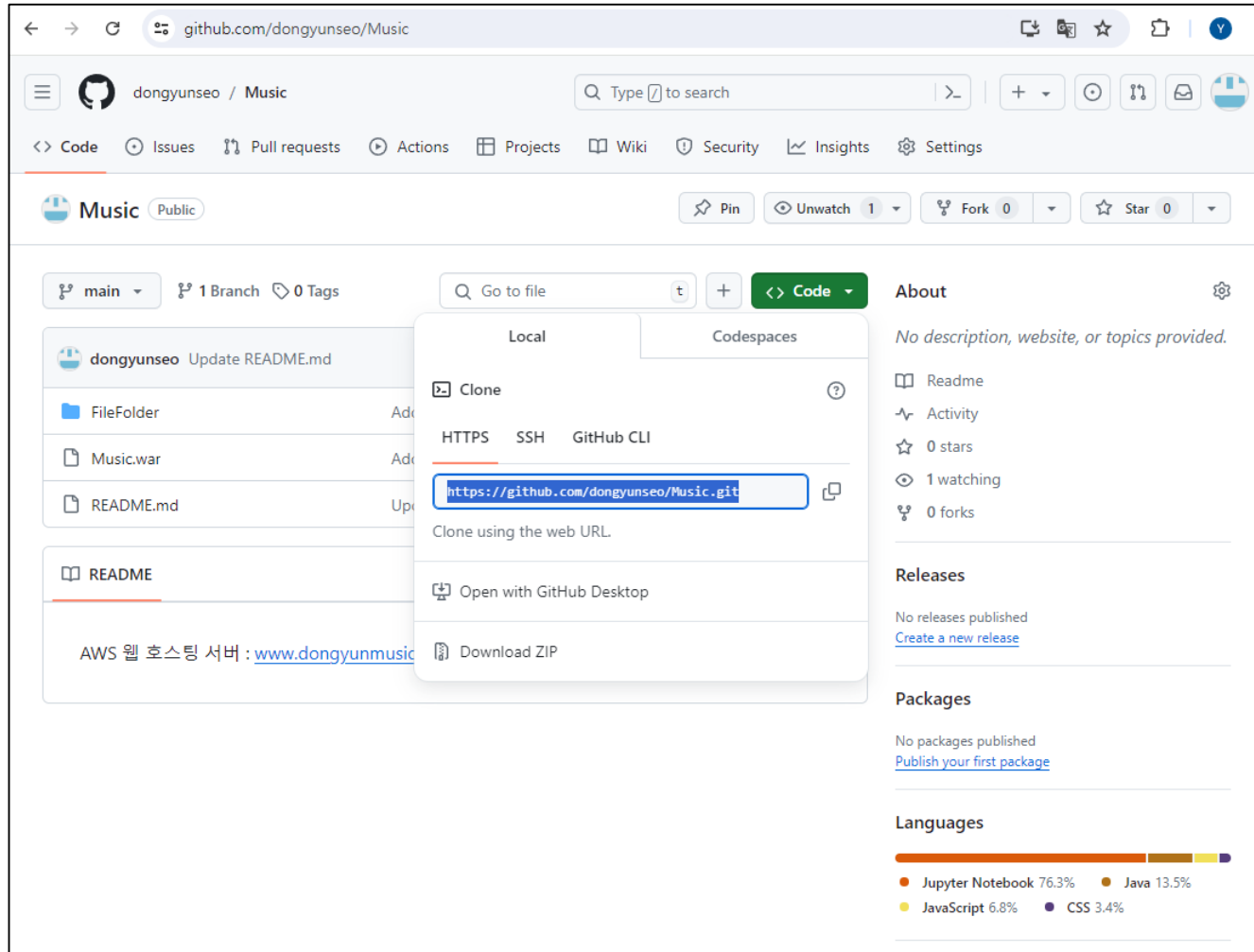
※ 다운로드한 파일 압축해제 명령어 : tar -xvf tomcat.tar.gz

```
ubuntu@ip-172-31-35-166:~$ ls
apache-tomcat-10.1.13  tomcat.tar.gz
```

2. 플랫폼 별 음원 차트 서비스

3.1 AWS 서버 연결

3.1.15 JSP 파일 업로드 sudo git clone <https://github.com/dongyunseo/Music.git> 해당 명령어를 통해 war 파일 다운로드



```
ubuntu@ip-172-31-35-166:~/apache-tomcat-9.0.87/webapps$ ls
Music Music.war ROOT docs examples host-manager manager
ubuntu@ip-172-31-35-166:~/apache-tomcat-9.0.87/webapps$
```

2. 플랫폼 별 음원 차트 서비스

3.1 AWS 서버 연결

3.1.16 해당 경로에 /startup.sh 명령어를 통해 실행시켜줍니다

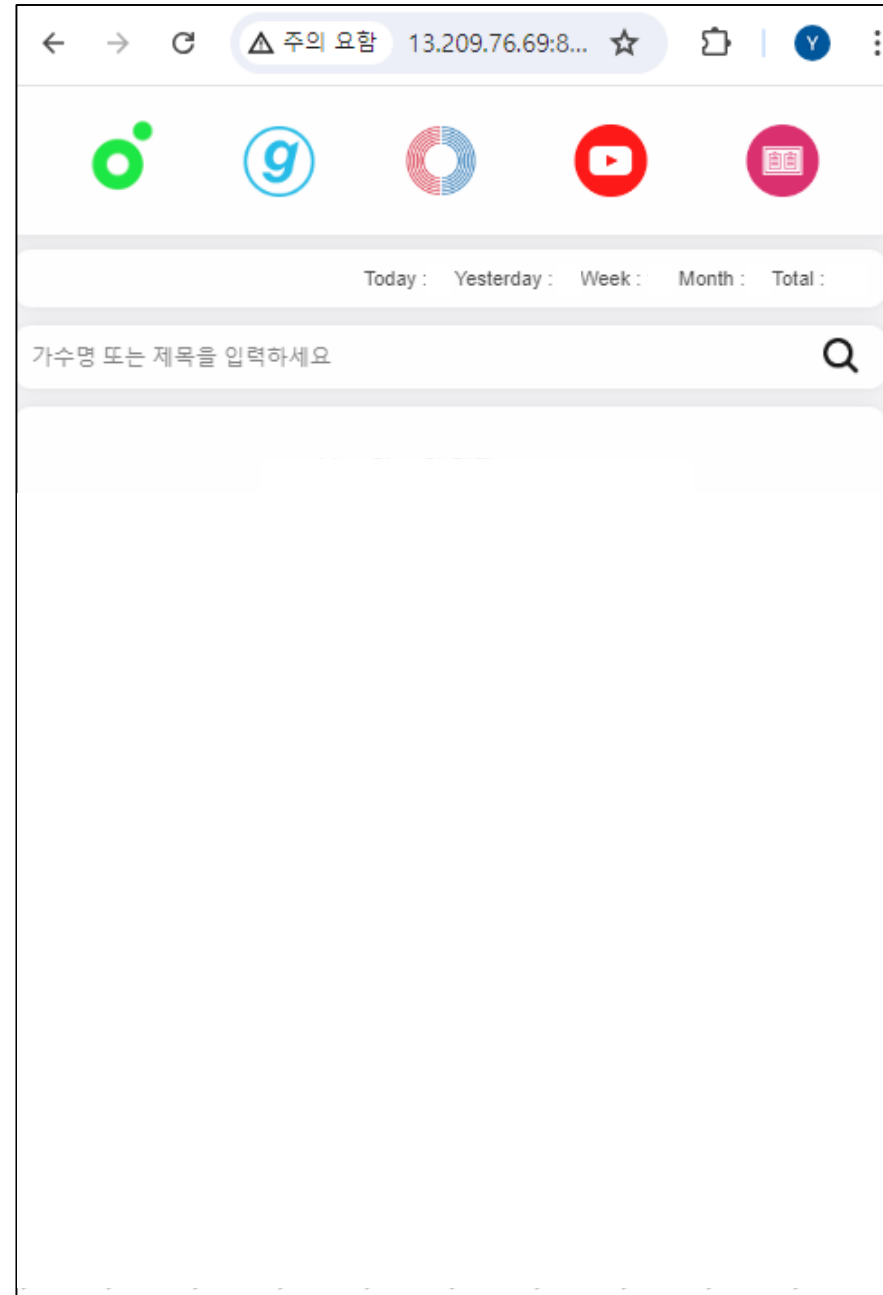
```
ubuntu@ip-172-31-35-166:~/apache-tomcat-9.0.87/bin$ ls
bootstrap.jar      ciphers.bat      configtest.bat  digest.sh        setclasspath.sh  startup.sh        tool-wrapper.sh
catalina-tasks.xml ciphers.sh       configtest.sh   makebase.bat     shutdown.bat      tomcat-juli.jar   version.bat
catalina.bat       commons-daemon-native.tar.gz daemon.sh        makebase.sh      shutdown.sh       tomcat-native.tar.gz version.sh
catalina.sh        commons-daemon.jar digest.bat       setclasspath.bat startup.bat        tool-wrapper.bat
ubuntu@ip-172-31-35-166:~/apache-tomcat-9.0.87/bin$
```

```
ubuntu@ip-172-31-35-166:/apache-tomcat-9.0.87/bin$ ./startup.sh
Using CATALINA_BASE:   /home/ubuntu/apache-tomcat-9.0.87
Using CATALINA_HOME:   /home/ubuntu/apache-tomcat-9.0.87
Using CATALINA_TMPDIR: /home/ubuntu/apache-tomcat-9.0.87/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/ubuntu/apache-tomcat-9.0.87/bin/bootstrap.jar
13/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
```


2. 플랫폼 별 음원 차트 서비스

3.1 AWS 서버 연결

3.1.17 aws 인스턴스 퍼블릭 IPv4 주소를 입력 후 폴더 위치와 함께 입력하여 접속



2. 플랫폼 별 음원 차트 서비스

3.1 AWS 서버 연결

3.1.18 Mysql 설치 : apt-cache search mysql | grep 8.0 해당 명령어로 패키지 리스트가 있는지 확인 후 설치

3.1.19 설치 후 netstat -nlpt 명령어를 통해 3306포트가 뜨면 정상적으로 설치완료

```
ubuntu@ip-172-31-35-166:~$ sudo apt-get install mysql-server-8.0
```

```
ubuntu@ip-172-31-35-166:~$ netstat -nlpt
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:33060        0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:3306         0.0.0.0:*               LISTEN      -
tcp6       0      0 127.0.0.1:8005         :::*                   LISTEN      66849/java
tcp6       0      0 :::22                  :::*                   LISTEN      -
tcp6       0      0 :::8080                 :::*                   LISTEN      66849/java
ubuntu@ip-172-31-35-166:~$
```

2. 플랫폼 별 음원 차트 서비스

3.1 AWS 서버 연결

3.1.20 sudo mysql -u root -p를 입력하고 비밀번호를 입력하면 Mysql 접속가능

```
ubuntu@ip-172-31-35-166:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4619
Server version: 8.0.36-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

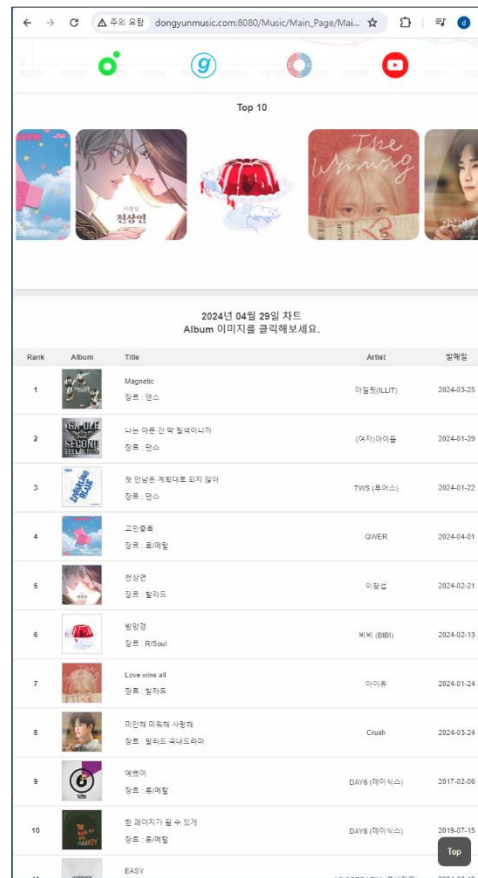
mysql> USE Music
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
```

2. 플랫폼 별 음원 차트 서비스

3.1 AWS 서버 연결

3.1.21 해당 테이블 및 데이터 삽입 후 `sudo service mysql restart` 명령어를 통해 Mysql을 재부팅해주면 정상적으로 화면에 출력됨을 확인



가수명 또는 제목을 입력하세요

Number	Img	Title	Singer
1		사랑하지 않아서 그랬니	지아
2		사랑하지 않아서 그랬니	지아
3		사랑하지 않아서 그랬니	지아
4		사랑하지 않아서 그랬니	지아
5	
6	
7	
8	
9	
10	

감사합니다.

1. 쇼핑몰 Git : https://github.com/dongyunseo/shopping_mall_1215
2. 플랫폼 별 음원 차트 서비스 Git : <https://github.com/dongyunseo/Music>
3. 콘텐츠 기반 추천시스템 개발 Git : https://github.com/dongyunseo/Python_Project