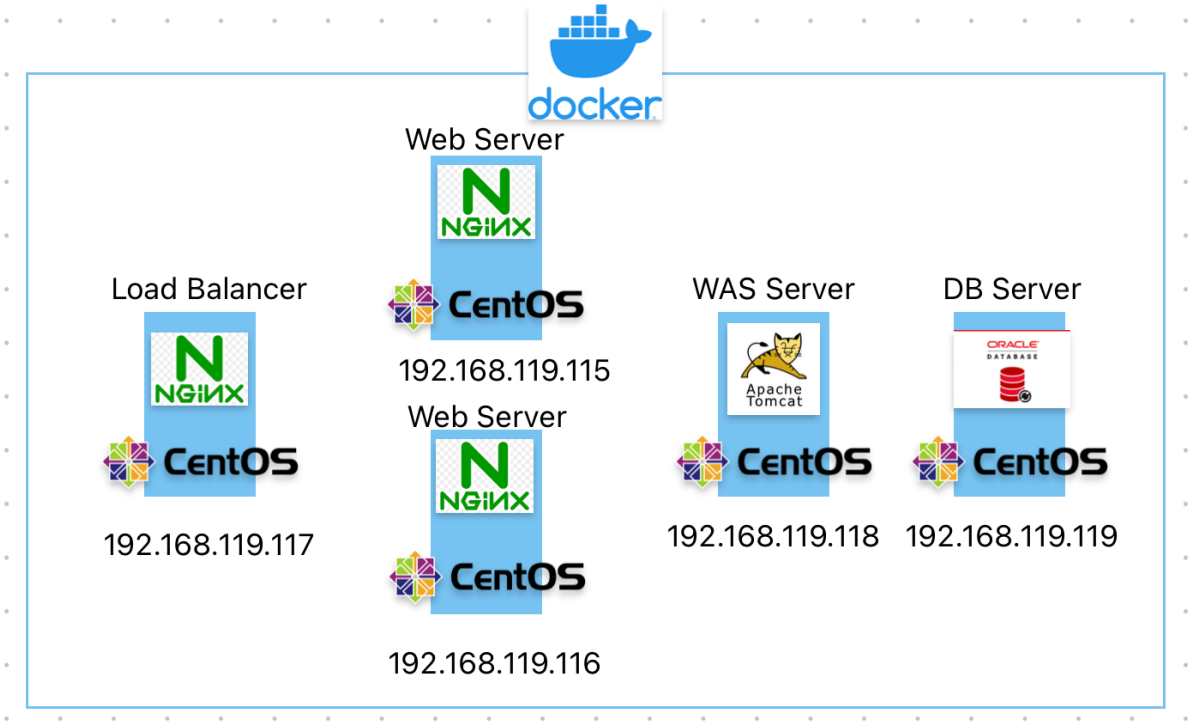
6. 서버연동

구조



WEB<->WAS 연동

nginx.conf 수정

```
nano /etc/nginx/nginx.conf
```

upstream 설정

```
upstream tomcat {
        server 192.168.119.118:8080;
        keepalive 32;
    }
```

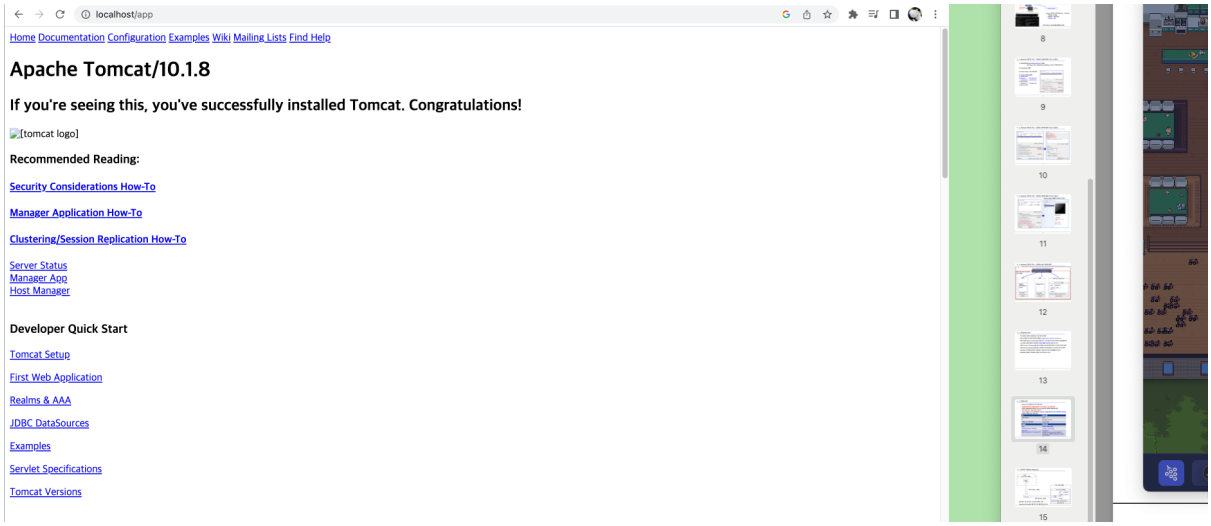
url 디렉터리 설정

```
location / {
                root /usr/share/nginx/html;
                index index.html;
        }

location /app {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
```

```
            proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
        proxy_set_header X-NginX-Proxy true;

        proxy_pass http://tomcat/;
        proxy_redirect off;
    }
```

재시동

```
sudo systemctl restart nginx
```



추가 파일 넣어주기

WASSVR 컨테이너에서 WEBSVR로 필요한 파일을 전송한다.

WASSVR에서 실행

```
cd /opt/tomcat/webapps/ROOT
```

scp를 사용하기 위해 openssh-clients설치

```
yum install openssh-clients
```

```
Running transaction
  Installing : fipscheck-1.4.1-6.el7.x86_64                                 1/5
  Installing : fipscheck-lib-1.4.1-6.el7.x86_64                             2/5
  Installing : openssh-7.4p1-22.el7_9.x86_64                                3/5
  Installing : libedit-3.0-12.20121213cvs.el7.x86_64                        4/5
  Installing : openssh-clients-7.4p1-22.el7_9.x86_64                        5/5
  Verifying  : fipscheck-lib-1.4.1-6.el7.x86_64                             1/5
  Verifying  : openssh-7.4p1-22.el7_9.x86_64                                2/5
  Verifying  : fipscheck-1.4.1-6.el7.x86_64                                 3/5
  Verifying  : libedit-3.0-12.20121213cvs.el7.x86_64                        4/5
  Verifying  : openssh-clients-7.4p1-22.el7_9.x86_64                        5/5

Installed:
  openssh-clients.x86_64 0:7.4p1-22.el7_9

Dependency Installed:
  fipscheck.x86_64 0:1.4.1-6.el7            fipscheck-lib.x86_64 0:1.4.1-6.el7
  libedit.x86_64 0:3.0-12.20121213cvs.el7   openssh.x86_64 0:7.4p1-22.el7_9

Complete!
```

```
scp tomcat.svg root@192.168.119.117:/usr/share/nginx/html/
```

```
[root@da2d98182168 ROOT]# scp tomcat.svg root@192.168.119.117:/usr/share/nginx/h
tml/
ssh: connect to host 192.168.119.117 port 22: Connection refused
lost connection
```

컨테이너 만들때 22포트를 열어두지 않았다. 그냥 컨테이너 → 로컬 → 컨테이너 방식을 사용해야한다.

도커의 파일 가져오기

```
docker cp WASSVR:/opt/tomcat/webapps/ROOT/tomcat.svg .
docker cp WASSVR:/opt/tomcat/webapps/ROOT/tomcat.css .
```

```
Hong-YoonKi@Hong-YoonKi-MacBookAir ~ % docker cp WASSVR:/opt/tomcat/webapps/ROOT/tomcat.svg .
                                        Successfully copied 69.6kB to /Users/Hong-YoonKi/.
Hong-YoonKi@Hong-YoonKi-MacBookAir ~ % docker cp WASSVR:/opt/tomcat/webapps/ROOT/tomcat.css .
                                        Successfully copied 7.17kB to /Users/Hong-YoonKi/.
Hong-YoonKi@Hong-YoonKi-MacBookAir ~ % ls
[redacted]                                                                  [redacted]
                                                                            tomcat.css
                                                                            tomcat.svg
[redacted]                                                                  [redacted]
```

가져오기 성공

도커로 파일 내보내기

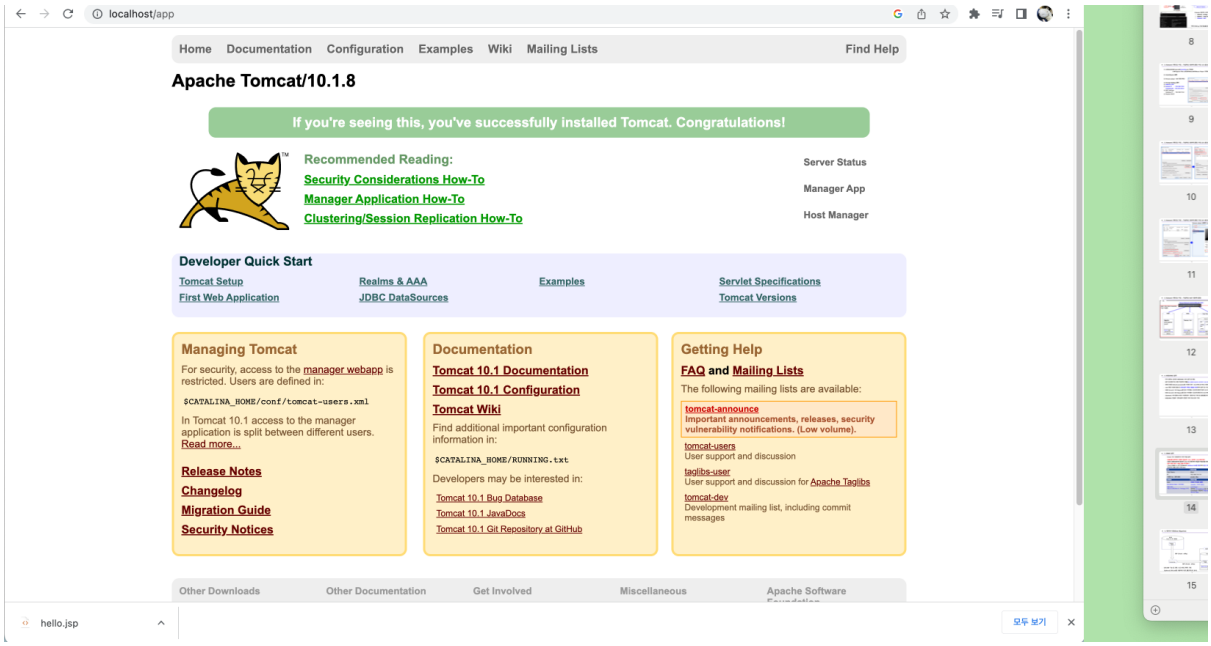
```
docker cp tomcat.svg WEBSVR:/usr/share/nginx/html/
docker cp tomcat.css WEBSVR:/usr/share/nginx/html/
```

```
Hong-YoonKi@Hong-YoonKi-MacBookAir ~ % docker cp tomcat.svg WEBSVR:/usr/share/nginx/html/
                                Successfully copied 69.6kB to WEBSVR:/usr/share/nginx/html/
Hong-YoonKi@Hong-YoonKi-MacBookAir ~ % docker cp tomcat.css WEBSVR:/usr/share/nginx/html/
                                Successfully copied 7.17kB to WEBSVR:/usr/share/nginx/html/
Hong-YoonKi@Hong-YoonKi-MacBookAir ~ % docker exec -it WEBSVR /bin/bash
[root@b6d81fdfa1b1 /]# cd /usr/share/nginx/html/
[root@b6d81fdfa1b1 html]# ls
404.html  50x.html  en-US  icons  img  index.html  nginx-logo.png  poweredby.png  tomcat.css  tomcat.svg
```

권한 부여

```
chown -R nginx:nginx /usr/share/nginx/html/
chown nginx:nginx /usr/share/nginx/html/tomcat.css
chown nginx:nginx /usr/share/nginx/html/tomcat.svg
```

css와 svg에 대한 접근 권한을 주어야 접근이 가능해 깨지지 않고 나온다.



WAS <-> DBMS 연동

odbc 복사

```
docker cp ojdbc11.jar WASSVR:/opt/tomcat/lib
```

```
[root@da2d98182168 lib]# pwd
/opt/tomcat/lib
[root@da2d98182168 lib]# ls
annotations-api.jar                    tomcat-dbcp.jar
catalina-ant.jar                       tomcat-i18n-cs.jar
catalina-ha.jar                        tomcat-i18n-de.jar
catalina.jar                           tomcat-i18n-es.jar
catalina-ssi.jar                       tomcat-i18n-fr.jar
catalina-storeconfig.jar               tomcat-i18n-ja.jar
catalina-tribes.jar                    tomcat-i18n-ko.jar
ecj-4.27.jar                           tomcat-i18n-pt-BR.jar
el-api.jar                             tomcat-i18n-ru.jar
jakartaee-migration-1.0.6-shaded.jar   tomcat-i18n-zh-CN.jar
jasper-el.jar                          tomcat-jdbc.jar
jasper.jar                             tomcat-jni.jar
jaspic-api.jar                         tomcat-util.jar
jsp-api.jar                            tomcat-util-scan.jar
ojdbc11.jar                            tomcat-websocket.jar
servlet-api.jar                        websocket-api.jar
tomcat-api.jar                         websocket-client-api.jar
tomcat-coyote.jar
```

jdbc 연동 확인

확인용 디렉터리 생성

```
mkdir jdbcTest
```

context.xml 수정

```
nano /opt/tomcat/conf/context.xml
```

```
<Resource name="jdbc/dink22" auth="Container" type="javax.sql.DataSource"
        maxTotal="100" maxIdle="30" maxWaitMillis="10000"
        username="C##SCOTT" password="tiger"
        driverClassName="oracle.jdbc.driver.OracleDriver"
        url="jdbc:oracle:thin:@192.168.119.119:1521:dink22"
        factory="org.apache.tomcat.jdbc.pool.DataSourceFactory"
        testWhileIdle="true" validationQuery="SELECT 1 FROM DUAL"/>
```

변경사항 반영

```
sudo systemctl daemon-reload
sudo systemctl start tomcat
sudo systemctl enable tomcat
```

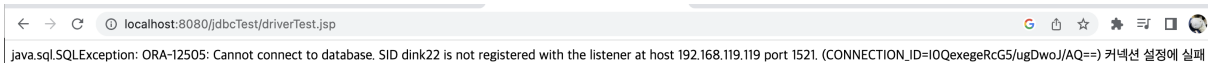
Test jsp 전송

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

<%
  Connection conn = null;

  try{
    Class.forName("oracle.jdbc.driver.OracleDriver");
  }catch(ClassNotFoundException cnfe){
    cnfe.printStackTrace();
    System.out.println("드라이버 로딩 실패");
  }
  try{
    String jdbcUrl = "jdbc:oracle:thin:@192.168.119.119:1521:dink22";
    String userId = "C##SCOTT";
    String userPass = "tiger";
    conn = DriverManager.getConnection(jdbcUrl, userId, userPass);
    out.println("접속 성공");
  }catch(SQLException e){
    e.printStackTrace();
    out.println(e);
    out.println("커넥션 설정에 실패");
  }
%>
```

```
docker cp driverTest.jsp WASSVR:/opt/tomcat/webapps/ROOT/jdbcTest
docker cp hellosql.jsp WASSVR:/opt/tomcat/webapps/ROOT/jdbcTest
```

```
[root@da2d98182168 jdbcTest]# ls
driverTest.jsp  hellosql.jsp
```

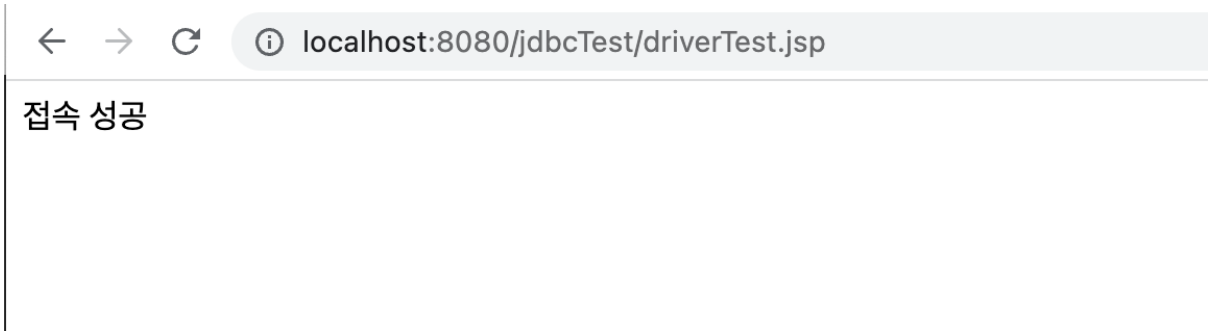
Test jsp 작동 확인



드라이버 로딩에 성공한 듯 하지만, 커넥션을 설정하는 것에 실패했다.

```
[oracle@cf6c9a691f17 /]$ sqlplus sys/enter#123 as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Sat May 6 11:47:34 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup
;
ORACLE instance started.

Total System Global Area 1577055360 bytes
Fixed Size                  9135232 bytes
Variable Size             436207616 bytes
Database Buffers         1124073472 bytes
Redo Buffers                7639040 bytes
Database mounted.
Database opened.
```

DB를 켜지 않았었다.



DB켜니 성공한다.

다른 jsp로도 테스트해보자

```
# name = hellosql.jsp
<%@ page import="java.sql.*" %>
<%@ page import="javax.naming.*, javax.sql.*" %>
<%@ page import="java.util.*" %>
<%@ page import="java.io.*" %>

<html>
<head>
    <title>Hello JSP</title>
</head>
<body>
    <h1>Employee List</h1>

    <%-- JDBC 연결 정보 --%>
    <%!
        private Connection getConnection() throws Exception {
            Context initCtx = new InitialContext();
            DataSource ds = (DataSource) initCtx.lookup("java:/comp/env/jdbc/dink22");
            Connection conn = ds.getConnection();
            return conn;
```

```
        }
    %>

    <%-- 데이터베이스 연결 및 쿼리 실행 --%>
    <%
        Connection conn = null;
        Statement stmt = null;
        ResultSet rs = null;

        try {
            conn = getConnection();
            stmt = conn.createStatement();
            rs = stmt.executeQuery("SELECT * FROM emp");
    %>

    <%-- 쿼리 결과 표시 --%>
    <table border="1">
        <tr>
            <th>EMPNO</th>
            <th>ENAME</th>
            <th>JOB</th>
            <th>DEPTNO</th>
        </tr>
        <% while (rs.next()) { %>
            <tr>
                <td><%= rs.getInt("EMPNO") %></td>
                <td><%= rs.getString("ENAME") %></td>
                <td><%= rs.getString("JOB") %></td>
                <td><%= rs.getInt("DEPTNO") %></td>
            </tr>
        <% } %>
    </table>

    <%-- 예외 처리 --%>
    <%
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (rs != null) rs.close();
            if (stmt != null) stmt.close();
            if (conn != null) conn.close();
        }
    %>

</body>
</html>
```

localhost/app/jdbcTest/hellosql.jsp

Employee List

EMPNO	ENAME	JOB	DEPTNO
7369	SMITH	CLERK	20
7499	ALLEN	SALESMAN	30
7521	WARD	SALESMAN	30
7566	JONES	MANAGER	20
7654	MARTIN	SALESMAN	30
7698	BLAKE	MANAGER	30
7782	CLARK	MANAGER	10
7788	SCOTT	ANALYST	20
7839	KING	PRESIDENT	10
7844	TURNER	SALESMAN	30
7876	ADAMS	CLERK	20
7900	JAMES	CLERK	30
7902	FORD	ANALYST	20
7934	MILLER	CLERK	10

web-was-DB연동에 성공한 것을 확인 가능하다.

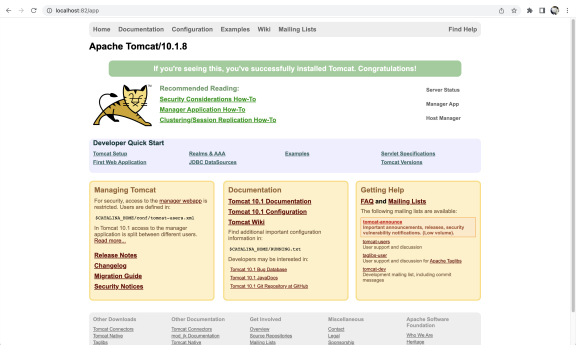
Load Balancer

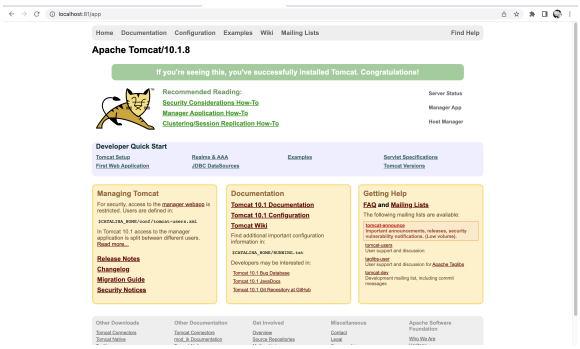
웹서버 2개 생성

```
docker run --privileged -d --name WEBSVR2 -p 81:80 --net mynet --ip 192.168.119.116 websvrimg /sbin/init
docker run --privileged -d --name WEBSVR1 -p 82:80 --net mynet --ip 192.168.119.115 websvrimg /sbin/init
```

기존 웹서버 이름 변경 → load balancer

```
docker rename WEBSVR LDBLNCR
```





웹서버 2개 완성 뚝딱

Load Balancing

/etc/nginx/nginx.conf

```
upstream tomcat {
        server 192.168.119.116:80;
        server 192.168.119.115:80;
    }
location / {
            proxy_pass http://tomcat/;
        }
```

```
Hong-YoonKi — @b6d81fdf
GNU nano 2.3.1                                                    File: nginx

    default_type        application/octet-stream;

    # Load modular configuration files from the /etc/nginx/conf.d directory.
    # See http://nginx.org/en/docs/ngx_core_module.html#include
    # for more information.
    include /etc/nginx/conf.d/*.conf;
    types {
        image/svg+xml svg;
        text/css css;
    }

    upstream tomcat {
        server 192.168.119.116:80;
        server 192.168.119.115:80;
    }

    server {
        listen       80;
        listen       [::]:80;
        server_name  _;

        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;

        location / {
            proxy_pass http://tomcat/;
        }

        error_page 404 /404.html;
        location = /404.html {
        }

        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
        }
    }
```

확인하기

localhost/app/jdbcTest/hellosql.jsp

Employee List

EMPNO	ENAME	JOB	DEPTNO
7369	SMITH	CLERK	20
7499	ALLEN	SALESMAN	30
7521	WARD	SALESMAN	30
7566	JONES	MANAGER	20
7654	MARTIN	SALESMAN	30
7698	BLAKE	MANAGER	30
7782	CLARK	MANAGER	10
7788	SCOTT	ANALYST	20
7839	KING	PRESIDENT	10
7844	TURNER	SALESMAN	30
7876	ADAMS	CLERK	20
7900	JAMES	CLERK	30
7902	FORD	ANALYST	20
7934	MILLER	CLERK	10

웹페이지에선 확인할 방법이 없다.

로그 파일을 작성하게 하여 LB가 잘 되는지 확인해 보자

/etc/nginx/nginx.conf

```
log_format upstream_log '$remote_addr - $remote_user [$time_local] '
                                '"$request" $status $body_bytes_sent '
                                '"$http_referer" "$http_user_agent" '
                                '$upstream_addr $upstream_status $upstream_response_time';
```

이걸 추가한다.

```
192.168.119.2 - - [08/May/2023:06:42:05 +0000] "GET /app/jdbcTest/hellosql.jsp HTTP/1.1" 200 2646 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36" "-"
192.168.119.2 - - [08/May/2023:06:51:42 +0000] "GET /app/jdbcTest/hellosql.jsp HTTP/1.1" 200 2646 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36" "-"192.168.119.116:80 200 0.010
192.168.119.2 - - [08/May/2023:06:51:43 +0000] "GET /app/jdbcTest/hellosql.jsp HTTP/1.1" 200 2646 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36" "-"192.168.119.115:80 200 0.060
192.168.119.2 - - [08/May/2023:06:51:47 +0000] "GET /app HTTP/1.1" 200 11235 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36" "-"192.168.119.116:80 200 0.000
192.168.119.2 - - [08/May/2023:06:51:47 +0000] "GET /asf-logo-wide.svg HTTP/1.1" 404 3650 "http://localhost/tomcat.css" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36" "-"192.168.119.115:80 404 0.010
192.168.119.2 - - [08/May/2023:06:51:47 +0000] "GET /bg-upper.png HTTP/1.1" 404 3650 "http://localhost/tomcat.css" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36" "-"192.168.119.116:80 404 0.000
192.168.119.2 - - [08/May/2023:06:51:47 +0000] "GET /bg-nav.png HTTP/1.1" 404 3650 "http://localhost/tomcat.css" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36" "-"192.168.119.116:80 404 0.010
192.168.119.2 - - [08/May/2023:06:51:47 +0000] "GET /bg-middle.png HTTP/1.1" 404 3650 "http://localhost/tomcat.css" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36" "-"192.168.119.115:80 404 0.000
192.168.119.2 - - [08/May/2023:06:51:47 +0000] "GET /bg-button.png HTTP/1.1" 404 3650 "http://localhost/tomcat.css" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36" "-"192.168.119.115:80 404 0.000
192.168.119.2 - - [08/May/2023:06:51:51 +0000] "GET /appjdbcTest/hellosql.jsp HTTP/1.1" 200 2646 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36" "-"192.168.119.116:80 200 0.070
192.168.119.2 - - [08/May/2023:06:53:48 +0000] "GET /appjdbcTest/hellosql.jsp HTTP/1.1" 200 2646 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36" "-"192.168.119.116:80 200 0.040
192.168.119.2 - - [08/May/2023:06:54:00 +0000] "GET /bg-middle.png HTTP/1.1" 404 3650 "http://localhost/tomcat.css" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36" "-"192.168.119.116:80 404 0.010
192.168.119.2 - - [08/May/2023:06:54:00 +0000] "GET /bg-nav.png HTTP/1.1" 404 3650 "http://localhost/tomcat.css" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36" "-"192.168.119.115:80 404 0.010
192.168.119.2 - - [08/May/2023:06:54:00 +0000] "GET /asf-logo-wide.svg HTTP/1.1" 404 3650 "http://localhost/tomcat.css" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36" "-"192.168.119.116:80 404 0.010
192.168.119.2 - - [08/May/2023:06:54:00 +0000] "GET /bg-button.png HTTP/1.1" 404 3650 "http://localhost/tomcat.css" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36" "-"192.168.119.115:80 404 0.010
192.168.119.2 - - [08/May/2023:06:54:00 +0000] "GET /bg-upper.png HTTP/1.1" 404 3650 "http://localhost/tomcat.css" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36" "-"192.168.119.115:80 404 0.010
192.168.119.2 - - [08/May/2023:06:54:00 +0000] "GET /app HTTP/1.1" 200 11235 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36" "-"192.168.119.115:80 200 0.130
192.168.119.2 - - [08/May/2023:06:54:07 +0000] "GET /app/hello.jsp HTTP/1.1" 200 231 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36" "-"192.168.119.116:80 200 0.040
```

성공적으로 이뤄진다.