

# Cloud Web Service-Based on Public share repository.

Team Name : Edelweiss

01

프로젝트 개요 및 일정.

## CONTENTS

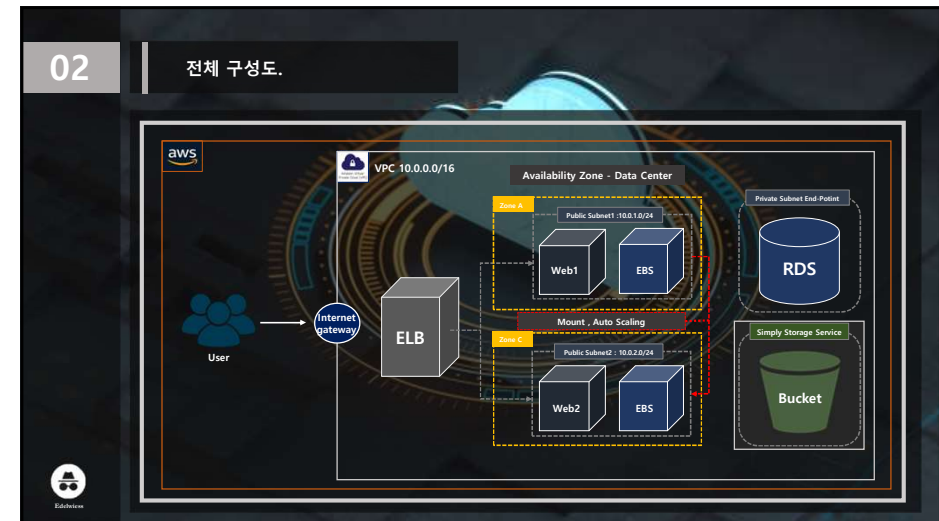
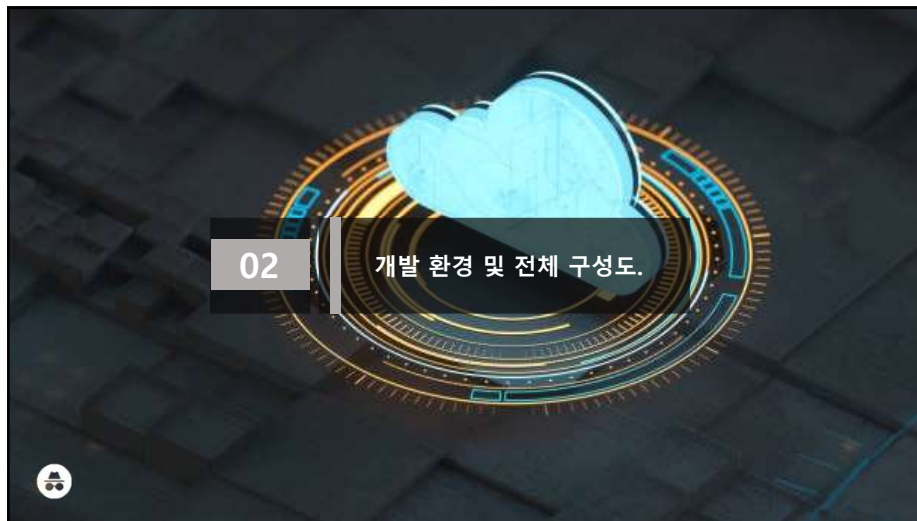
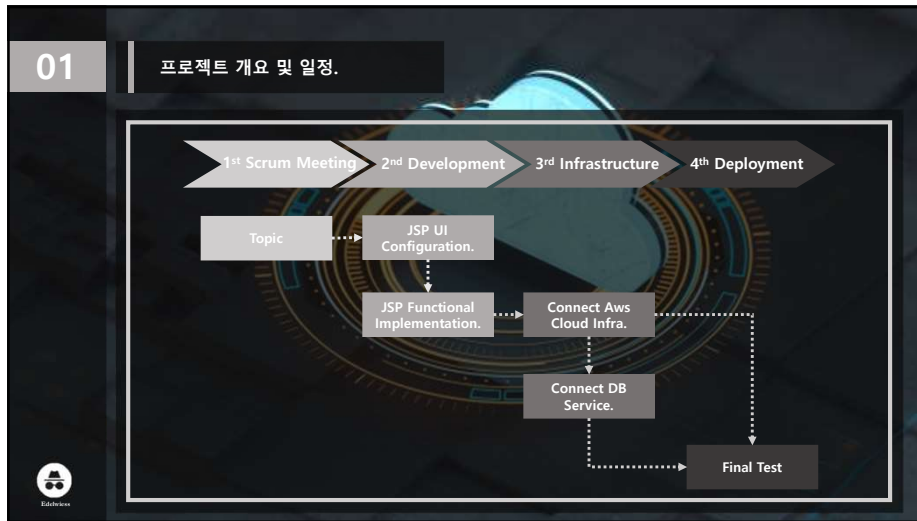
- 01 프로젝트 개요 및 일정.  
Project Overview and Schedule.
- 02 개발 환경 및 구성도.  
Development Environment and Configuration Diagram.
- 03 Aws 활용한 인프라 구축.  
Building Infrastructure Using Amazon Web Services.
- 04 JSP 구조 및 소스코드 설명.  
Description of the web development source code.
- 05 형상 관리.  
Configuration Management.
- 06 시연 영상.  
A demonstration video.

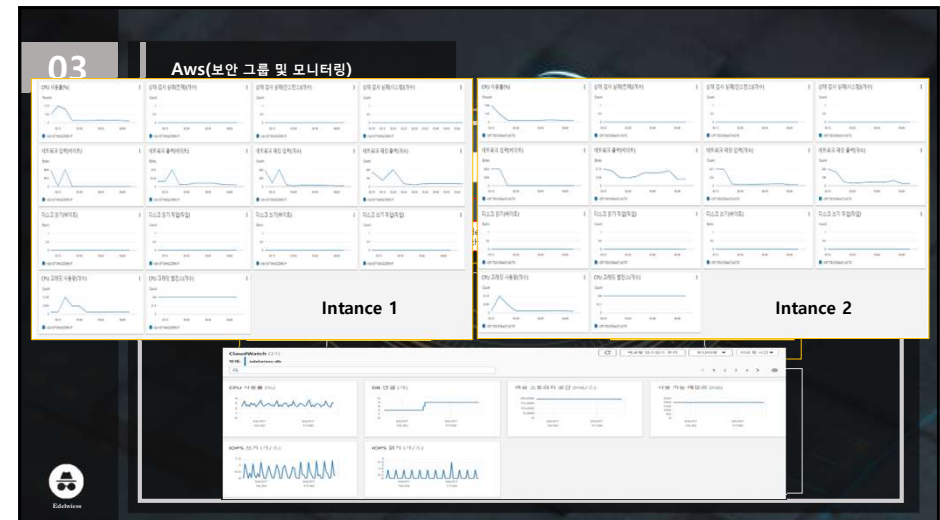
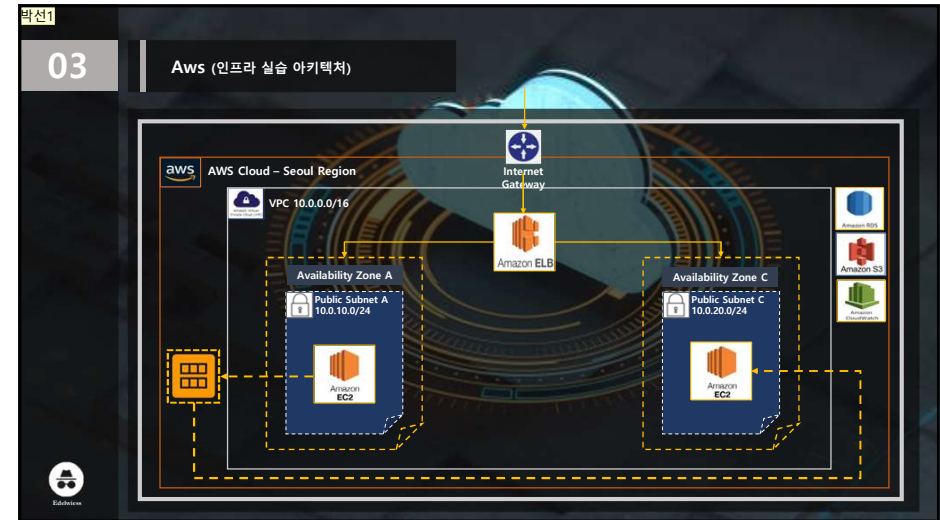
01

프로젝트 개요.

- 주제 : 아마존 웹 서비스를 이용한 공유 파일 저장소.
- 이름 : EPS (Edelweiss Public Share Storage)
- 개요 : 누구나 한번쯤 중요한 문서 나 파일, 이미지를 단톡방에 공유하거나 또는 자신에게 보내어 보관하는 경우가 있음.
- 이 때 무의식적으로 목적과 다른 채팅방에 보내는 문제점이 발생. 이를 해결하기 위해 팀원간 파일을 보관해 놓고 필요시 다룬 받을 수 있는 앱을 제작.











03

## AWS (MySQL DB 생성 및 테이블 생성)

```

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql> use edelwiess;
Database changed

mysql> select * from member;
+----+-----+-----+-----+-----+
| id | email | pwd | phone | message |
+----+-----+-----+-----+-----+
| 1 | test123 | test123 | 01011111111 | HELLO |
| 2 | test123 | test123 | 0101263841 | HELLO |
| 3 | test123 | test123 | 0101263841 | HELLO |
+----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> show tables;
+-----+
| Table |
+-----+
| file |
| member |
+-----+
2 rows in set (0.01 sec)

mysql> desc member;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| name | varchar(10) | YES | NULL | NULL | |
| email | varchar(50) | YES | PRIMARY | NULL | |
| pwd | varchar(12) | YES | NULL | NULL | |
| phone | varchar(13) | YES | NULL | NULL | |
| message | varchar(100) | YES | NULL | NULL | |
+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)

mysql> desc file;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| fileName | varchar(200) | YES | NULL | NULL | |
| fileRealName | varchar(200) | YES | NULL | NULL | |
+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> create table member (
  name varchar(10),
  email varchar(50) primary key,
  pwd varchar(12),
  phone varchar(13),
  message varchar(100)
);

mysql> create table file (
  fileName varchar(200),
  fileRealName varchar(200)
);

```

04

## JSP(Java Server Page)

03

## Mount Setting

- Server1 / server2  

```

ubuntu@server1:~$ vi /etc/hosts
127.0.0.1 localhost
3.36.103.204 server1 #Instance1
13.125.213.12 server2 #Instance2

```
- NFS를 사용하기 위한 server 설치  

```

ubuntu@server1:~$ apt install nfs-kernel-server

```
- NFS 설치  

```

ubuntu@server1:~$ apt install nfs-common

```
- 공유하기 위한 폴더 권한을 주어야 한다  

```

cd /var/lib/tomcat9/webapps/edelwiess/images/
chmod 777 upload

```
- 해당 폴더로 접근하고 사용하게 하기 위해서 권한을 지정해준다  

```

vi /etc/exports
/var/lib/tomcat9/webapps/edelwiess/images/upload
10.0.21.211(rw,sync,no_root_squash)

```
- nfs server 재부팅  

```

systemctl restart nfs-kernel-server

```
- 주소와 맵핑해준 hostname를 통해서 마운트  

```

mount -t nfs
server2:/var/lib/tomcat9/webapps/edelwiess/images/upload
/var/lib/tomcat9/webapps/edelwiess/images/upload

```
- 재부팅 시에도 마운트를 유지 시키기 위해서  

```

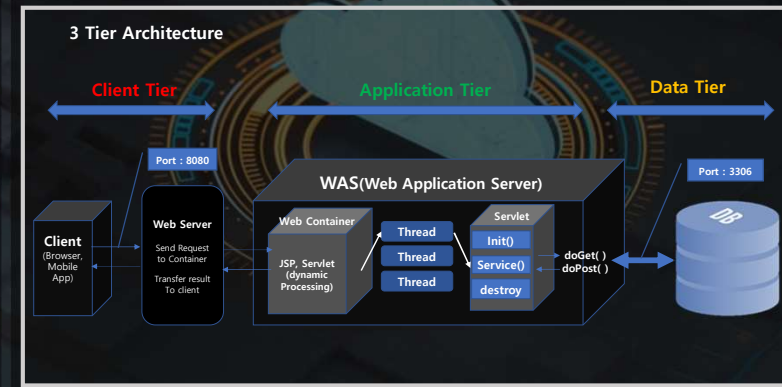
vi /etc/fstab
10.0.21.211:/var/lib/tomcat9/webapps/edelwiess/images/upload
/var/lib/tomcat9/webapps/edelwiess/images/upload
nfs defaults
0 0

```



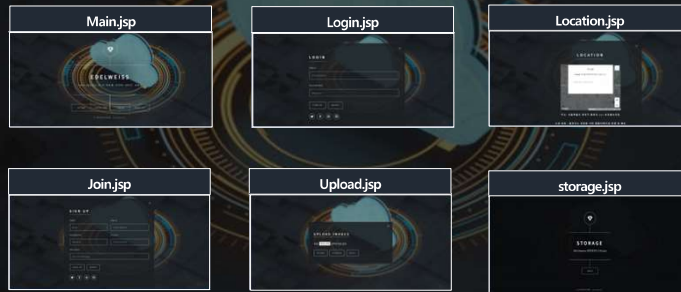
04

## JSP(Web Service Architecture)



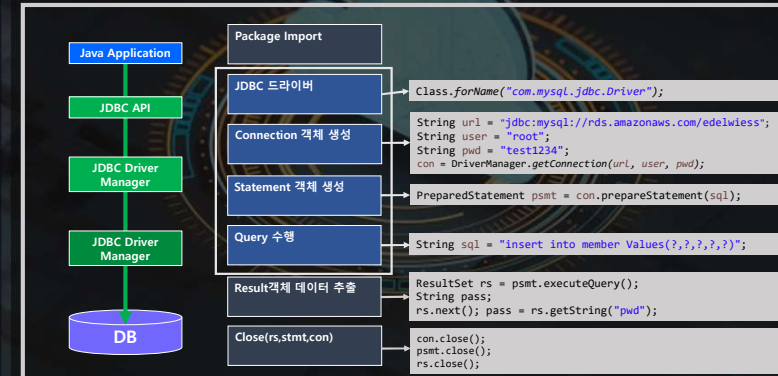
04

## JSP (프로젝트 전체 UI 화면)



04

## JDBC(Java Database Connectivity)



04

## JSP(주요 소스 코드 설명 DBUtil)

```

package DBUtil;
import java.sql.*;
import java.sql.DriverManager;

public class DBUtil {
    public static Connection getConnection() {
        Connection con = null;
        try {
            String url = "jdbc:mysql://edeliwies-db.ct7mvupkrjoh.ap-northeast-2-rds.amazonaws.com/edeliwies";
            String user = "root";
            String pwd = "test1234";
            Class.forName("com.mysql.jdbc.Driver"); // 1. JDBC 드라이버 로드
            con = DriverManager.getConnection(url, user, pwd);
            return con;
        } catch (Exception e) {
            // TODO: handle exception
            e.printStackTrace();
        }
        return null;
    }
}

```

04

## 주요 소스 코드 설명 (userDTO, userDAO)

```

package user;
public class userDTO {
    String userName;
    String userEmail;
    String userPwD;
    String userPhone;
    String userMessage;
    public String getUserName() {
        return userName;
    }
    public void setUserName(String userName) {
        this.userName = userName;
    }
    public String getUserEmail() {
        return userEmail;
    }
    public void setUserEmail(String userEmail) {
        this.userEmail = userEmail;
    }
    public String getUserPwD() {
        return userPwD;
    }
    public void setUserPwD(String userPwD) {
        this.userPwD = userPwD;
    }
    public String getUserPhone() {
        return userPhone;
    }
    public void setUserPhone(String userPhone) {
        this.userPhone = userPhone;
    }
    public String userMessage() {
        return userMessage;
    }
    public void setUserMessage(String userMessage) {
        this.userMessage = userMessage;
    }
}

public class userDAO {
    public int join(String userName, String userEmail, String userPwD, String userPhone, String userMessage) {
        String sql = "insert into member Values(?, ?, ?, ?, ?)";
        try {
            Connection con = DBUtil.getConnection();
            PreparedStatement pstmt = con.prepareStatement(sql);
            pstmt.setString(1, userName);
            pstmt.setString(2, userEmail);
            pstmt.setString(3, userPwD);
            pstmt.setString(4, userPhone);
            pstmt.setString(5, userMessage);
            return pstmt.executeUpdate();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

04

## 주요 소스 코드 설명 (FileDAO, FileDTO)

```

// edelwiess
// JAX-WS Web Services
// Java Resources
// src
// DBUtil.java
// file
// downloadAction.java
// FileDAO.java
// FileDTO.java
// user
// userDAO.java
// userDTO.java
// Libraries
// build
// WebContent
// assets
// images
// META-INF
// WEB-INF
// lib
// all
// mysql-connector-java-8.0.16
// tomcat-dbcp.jar
// fileDownload.jsp
// joinCK.jsp
// Login1.html
// Login2.html
// LoginCheck.jsp
// LoginMain.jsp
// Logout.jsp
// Main.jsp
// uploadAction.jsp
// web.xml

public class FileDAO {
    private Connection con;
    public FileDAO() {
        try {
            String url = "jdbc:mysql://edelwiess-db.ct7mvupkjh.ap-northeast-2.rds.amazonaws.com/edelwiess";
            String user = "root";
            String pwd = "test1234";
            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection(url, user, pwd);
        } catch (Exception e) {
            // TODO: handle exception
            e.printStackTrace();
        }
    }

    public int upload(String fileName, String fileRealName) {
        String SQL = "insert into file values (?,?)";
        try {
            PreparedStatement pstmt = con.prepareStatement(SQL);
            pstmt.setString(1, fileName);
            pstmt.setString(2, fileRealName);
            return pstmt.executeUpdate();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return -1;
    }
}

package file;
public class FileDTO {
    String fileName;
    String fileRealName;
    public String getFileName() {
        return fileName;
    }
    public void setFileName(String fileName) {
        this.fileName = fileName;
    }
    public String getRealName() {
        return fileRealName;
    }
    public void setRealName(String fileRealName) {
        this.fileRealName = fileRealName;
    }
    public FileDTO(String fileName, String fileRealName) {
        super();
        this.fileName = fileName;
        this.fileRealName = fileRealName;
    }
}

```

04

## 주요 소스 코드 설명 (upload, download)

```

// edelwiess
// JAX-WS Web Services
// Java Resources
// src
// DBUtil.java
// file
// downloadAction.java
// FileDAO.java
// FileDTO.java
// user
// userDAO.java
// userDTO.java
// Libraries
// build
// WebContent
// assets
// images
// META-INF
// WEB-INF
// lib
// all
// mysql-connector-java-8.0.16
// tomcat-dbcp.jar
// fileDownload.jsp
// joinCK.jsp
// Login1.html
// Login2.html
// LoginCheck.jsp
// LoginMain.jsp
// Logout.jsp
// Main.jsp
// uploadAction.jsp
// web.xml

<%
String directory = "/var/lib/tomcat9/webapps/edelwiess/images/upload/"; // 디렉토리 = 파일 저장
폴더명
int maxSize = 1024 * 1024 * 100; // 파일 크기 : 최대 100MB
String encoding = "UTF-8"; // 인코딩 타입

/* 파일이 저장된 위치 */
MultipartRequest multipartRequest = new MultipartRequest(request, directory, maxSize, encoding,
    new DefaultFileNamePolicy());

String fileName = multipartRequest.getOriginalFileName("file"); /* 실제 저장된 파일 이름 */
String fileRealName = multipartRequest.getFileName("file"); /* 저장된 파일의 원래 이름 */

// 아래와 같은 확장자가 아닐경우 파일을 생성후 삭제한다.
if(!fileName.endsWith(".png") && !fileName.endsWith(".hwp") && !fileName.endsWith(".jpg")
    && !fileName.endsWith(".hwp") && !fileName.endsWith(".png") && !fileName.endsWith(".ppt")
    && !fileName.endsWith(".xls")) {
    File file = new File(directory + fileName);
    file.delete();
}

/* 확장자 지정 경우 파일을 업로드한다.
else {
    new FileDAO().upload(fileName, fileRealName);
    out.write("<a href = \"" + request.getContextPath() + "/downloadAction?file="
        + java.net.URLEncoder.encode(fileName, "UTF-8") + "&w=" + file + "</a><br>");
}
%>

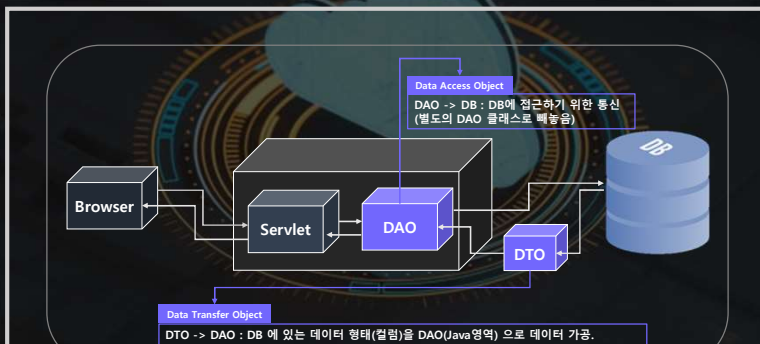
<%
String directory = "/var/lib/tomcat9/webapps/edelwiess/images/upload/";
String fileList = new File(directory).list();

for(String file : fileList) {
    out.write("<a href = \"" + request.getContextPath() + "/downloadAction?file="
        + java.net.URLEncoder.encode(fileName, "UTF-8") + "&w=" + file + "</a><br>");
}
%>

```

04

## DAO, DTO 소개.



04

## 주요 소스 코드 설명

```

// edelwiess
// JAX-WS Web Services
// Java Resources
// src
// DBUtil.java
// file
// downloadAction.java
// FileDAO.java
// FileDTO.java
// user
// userDAO.java
// userDTO.java
// Libraries
// build
// WebContent
// assets
// images
// META-INF
// WEB-INF
// lib
// all
// mysql-connector-java-8.0.16
// tomcat-dbcp.jar
// fileDownload.jsp
// joinCK.jsp
// Login1.html
// Login2.html
// LoginCheck.jsp
// LoginMain.jsp
// Logout.jsp
// Main.jsp
// uploadAction.jsp
// web.xml

<form class="form-signin" id="loginCheck" method="post">
    <input type="email" class="form-control" placeholder="Email address" name="inputEmail" required autofocus>
    </div>
    <div class="field half">
        <input type="password" class="form-control" placeholder="Password" name="inputPassword" required>
    </div>
    <div class="actions">
        <input type="submit" value="Sign in" />
        <input type="reset" value="Reset" />
    </div>
</form>

request.setCharacterEncoding("UTF-8");
String id = request.getParameter("inputEmail");
String pwd = request.getParameter("inputPassword");

Connection con = DBUtil.getConnection();

String sql = "select * from member where email=?";
PreparedStatement pstmt = con.prepareStatement(sql);
pstmt.setString(1, id);
ResultSet rs = pstmt.executeQuery();
String pass = rs.getString("pwd");

if(pass.equals(pwd)) { /* pwd를 기준으로 */
    out.println("<script>");
    out.println("alert('로그인 되었습니다.');
```





[illegible]

