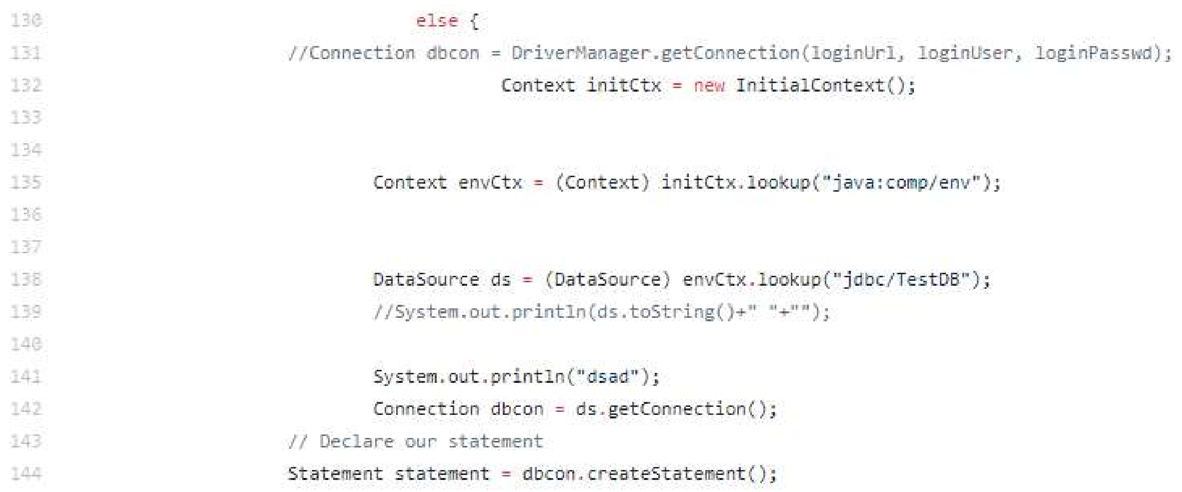
Task 1

- How did you use connection pooling?

  Context initCtx = new InitialContext();
  Context envCtx = (Context) initCtx.lookup("java:comp/env");
  DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB");
  Connection dbcon = ds.getConnection();

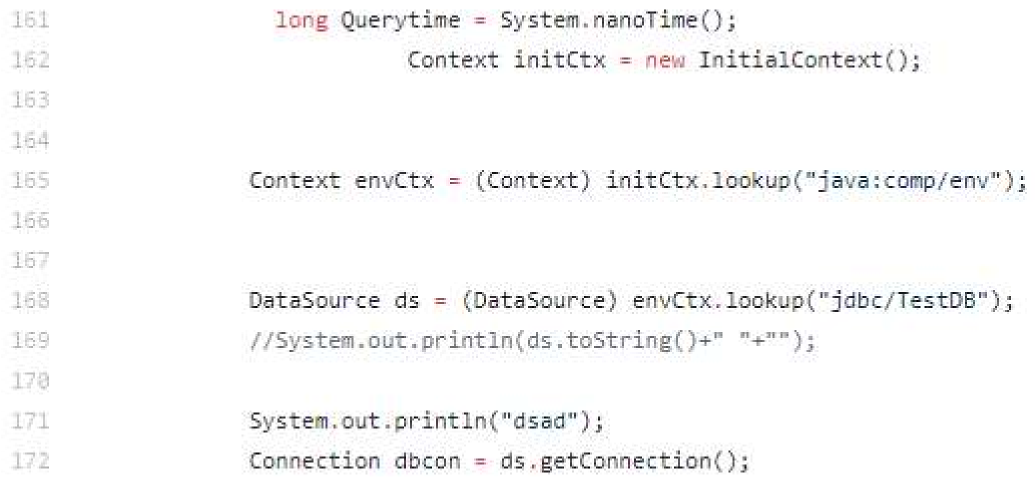
  Basically, what we did was comment out the previous user name and instance of connecting JDBC and added codes above. Which says that it will search details about jdbc from META-INF/context.xml and the name of the source is "jdbc/TestDB"

  - I Used pooling in login servlet, and other all servlet that requires a connection with jdbc. We had to copy the jar of connector is tomcat's global library, and restart tomcat.

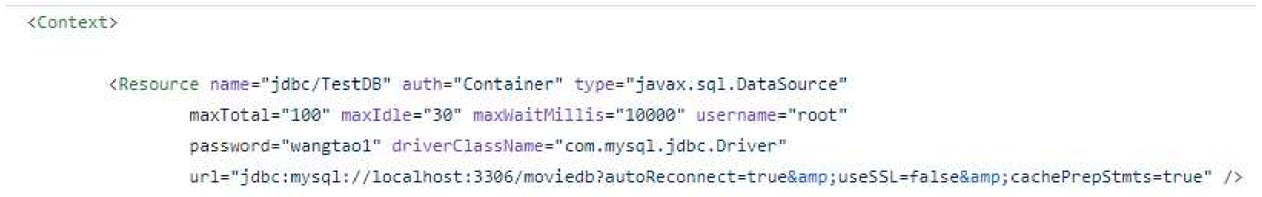


```java
                    else {
            //Connection dbcon = DriverManager.getConnection(loginUrl, loginUser, loginPasswd);
                            Context initCtx = new InitialContext();


                    Context envCtx = (Context) initCtx.lookup("java:comp/env");


                    DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB");
                    //System.out.println(ds.toString()+" "+"");

                    System.out.println("dsad");
                    Connection dbcon = ds.getConnection();
            // Declare our statement
            Statement statement = dbcon.createStatement();
```

The snapshot above is on normalsearch.java

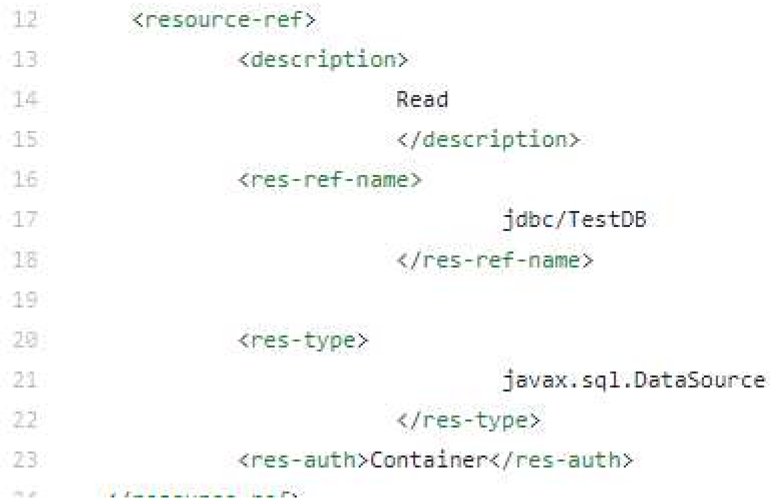


```java
            long Querytime = System.nanoTime();
                    Context initCtx = new InitialContext();


        Context envCtx = (Context) initCtx.lookup("java:comp/env");


        DataSource ds = (DataSource) envCtx.lookup("jdbc/TestDB");
        //System.out.println(ds.toString()+" "+"");

        System.out.println("dsad");
        Connection dbcon = ds.getConnection();
```



```
<Context>

        <Resource name="jdbc/TestDB" auth="Container" type="javax.sql.DataSource"
                maxTotal="100" maxIdle="30" maxWaitMillis="10000" username="root"
                password="wangtao1" driverClassName="com.mysql.jdbc.Driver"
                url="jdbc:mysql://localhost:3306/moviedb?autoReconnect=true&amp;useSSL=false&amp;cachePrepStmts=true" />
```

Above is context html which is where every servlet uses to connect to mysql.
I added cachePrepstmts = true to cache the prepared statements.
Routing in web.xml



```
        <resource-ref>
                <description>
                        Read
                        </description>
                <res-ref-name>
                                jdbc/TestDB
                        </res-ref-name>

                <res-type>
                                javax.sql.DataSource
                        </res-type>
                <res-auth>Container</res-auth>
```

-

- Login.java line 134
- normalsearch.java line 161
- Singletitle.java line 73
- TomcatTest.java line 83
- Bygenre.java line 126
- Bytitle.java line 132
- Checkout.java line 83
- Dashboard.java line 77
- Genres.java line 49
- Search.java line 195
- Shoppingcart.java line 99
- Singlestar.java line 73
- Tempauto.java line 113

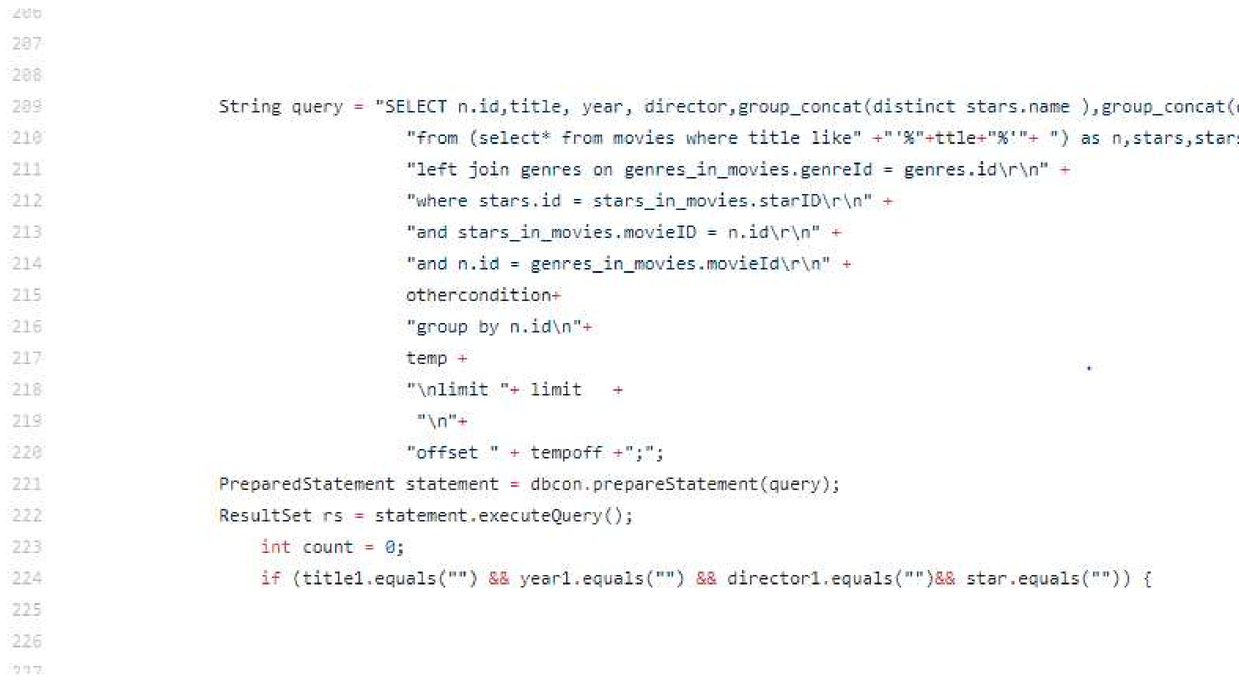
- How did you use Prepared Statements?

In the normal search or fulltext search, browsing by genre, and browsing by title, which are the searching parts, I used Preparestatement instead of createstatement. It will cache all the queries and send them at once. It will save workload of mysql server.

- Tempauto.java line 142
- Normalsearch.java line 210
- Search.java line 221

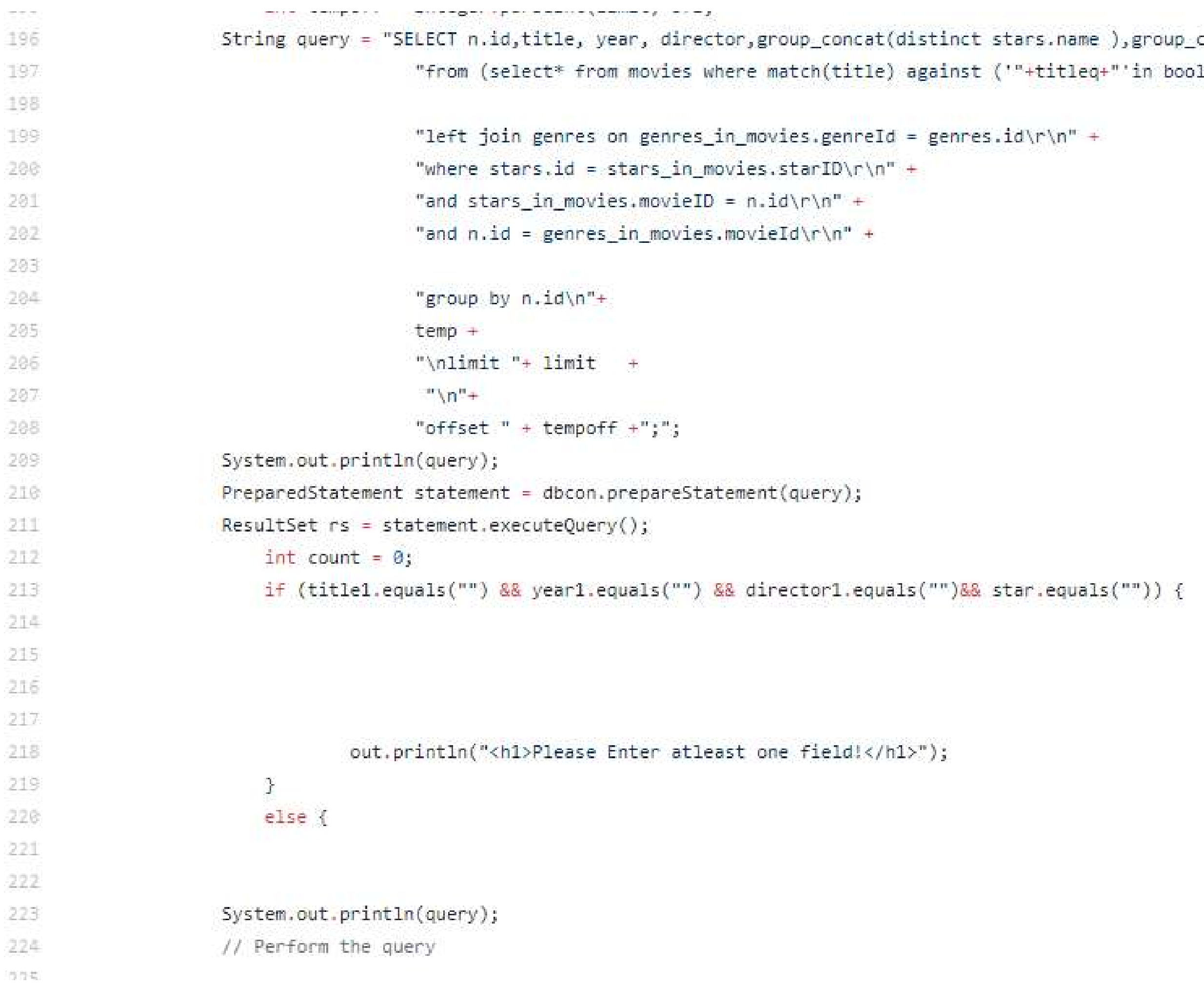
- Snapshots

-



```
206
207
208
209              String query = "SELECT n.id,title, year, director,group_concat(distinct stars.name ),group_concat(
210                                 "from (select* from movies where title like" +"'%"+ttle+"%'"+ ") as n,stars,star
211                                 "left join genres on genres_in_movies.genreId = genres.id\r\n" +
212                                 "where stars.id = stars_in_movies.starID\r\n" +
213                                 "and stars_in_movies.movieID = n.id\r\n" +
214                                 "and n.id = genres_in_movies.movieId\r\n" +
215                                 othercondition+
216                                 "group by n.id\n"+
217                                 temp +
218                                 "\nlimit "+ limit   +
219                                  "\n"+
220                                 "offset " + tempoff +";";
221              PreparedStatement statement = dbcon.prepareStatement(query);
222              ResultSet rs = statement.executeQuery();
223                  int count = 0;
224                  if (title1.equals("") && year1.equals("") && director1.equals("")&& star.equals("")) {
225
226
227
```

-



```
String query = "SELECT n.id,title, year, director,group_concat(distinct stars.name ),group_c
                "from (select* from movies where match(title) against ('"+titleq+"'in bool

                "left join genres on genres_in_movies.genreId = genres.id\r\n" +
                "where stars.id = stars_in_movies.starID\r\n" +
                "and stars_in_movies.movieID = n.id\r\n" +
                "and n.id = genres_in_movies.movieId\r\n" +

                "group by n.id\n"+
                temp +
                "\nlimit "+ limit   +
                 "\n"+
                "offset " + tempoff +";";
System.out.println(query);
PreparedStatement statement = dbcon.prepareStatement(query);
ResultSet rs = statement.executeQuery();
    int count = 0;
    if (title1.equals("") && year1.equals("") && director1.equals("")&& star.equals("")) {




            out.println("<h1>Please Enter atleast one field!</h1>");
    }
    else {


System.out.println(query);
// Perform the query
```

-

Task 2

- Address of AWS and Google instances
  http://13.58.198.173/ address of AWS load balancer
  http://35.193.198.180/ address of google cloud load balancer
  The google cloud one has only installed apache2, but AWS one installed everything, since it was the first instance we were using.
- Have you verified that they are accessible? Does Fablix site get opened both on Google's 80 port and AWS' 8080 port?
  All four instances are open. 80 for load balancers and 8080 for slave and master.
  Below are snapshots of two apaches, which will also means the slave and master are working properly.

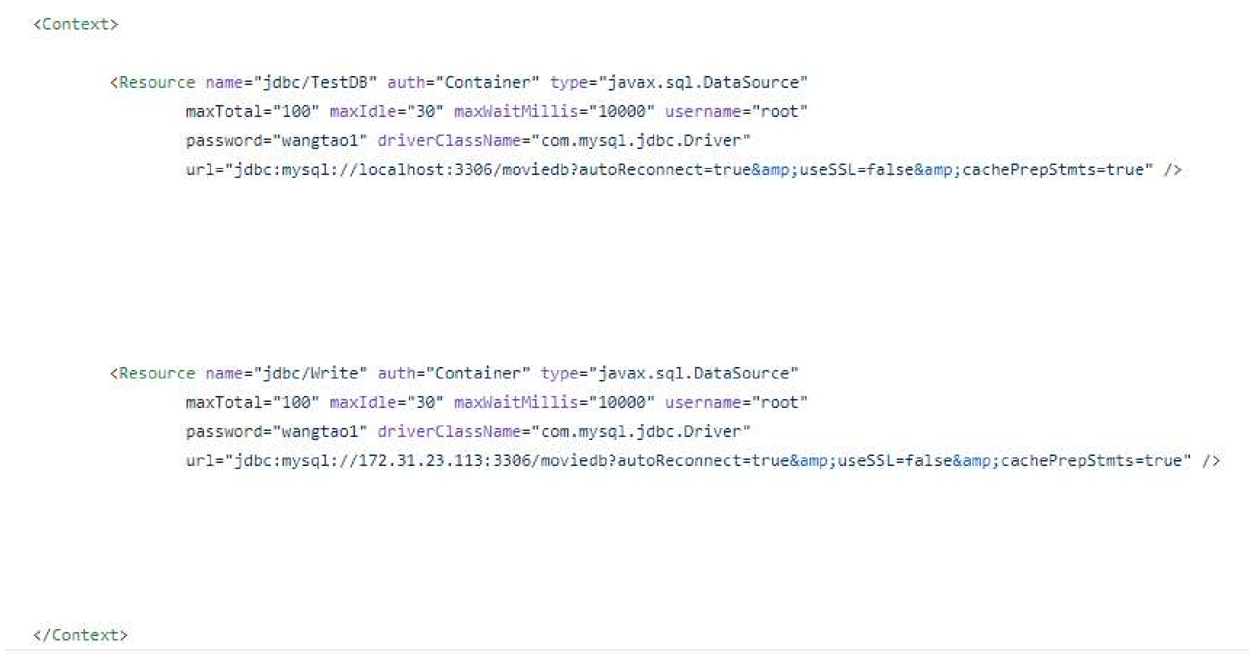
- How connection pooling works with two backend SQL?

I just added two resources in context.xml, and another routing in web.xml. One is localhost, which means when user is connected it will always go to its own mysql. The second resource also requires adding web.xml. To connect to master from slave we have to grant privilege to the slave in order to support remote control. Not doing so will cause an access denial from master.

- File name, line numbers as in Github
- In WebContent/META-INF/context.xml from line 1 to 22

- Snapshots

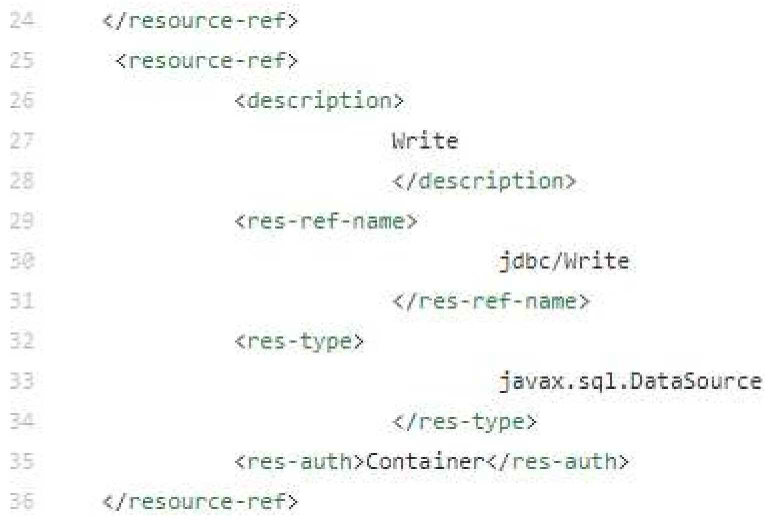
- Right here I used Write resource to connect to mysql server, since it is binded to master.
-



```
<Context>

        <Resource name="jdbc/TestDB" auth="Container" type="javax.sql.DataSource"
                maxTotal="100" maxIdle="30" maxWaitMillis="10000" username="root"
                password="wangtao1" driverClassName="com.mysql.jdbc.Driver"
                url="jdbc:mysql://localhost:3306/moviedb?autoReconnect=true&amp;useSSL=false&amp;cachePrepStmts=true" />



        <Resource name="jdbc/Write" auth="Container" type="javax.sql.DataSource"
                maxTotal="100" maxIdle="30" maxWaitMillis="10000" username="root"
                password="wangtao1" driverClassName="com.mysql.jdbc.Driver"
                url="jdbc:mysql://172.31.23.113:3306/moviedb?autoReconnect=true&amp;useSSL=false&amp;cachePrepStmts=true" />



</Context>
```

-

Web.xml is shown below.



```
    </resource-ref>
     <resource-ref>
             <description>
                         Write
                         </description>
             <res-ref-name>
                                 jdbc/Write
                         </res-ref-name>
             <res-type>
                                 javax.sql.DataSource
                         </res-type>
             <res-auth>Container</res-auth>
    </resource-ref>
```

- How read/write requests were routed?

Reading is always directed to the local host, which means it can be either master or slave. Writing will always directed to master's server, to maintain the consistency of slave master server.

- File name, line numbers as in Github

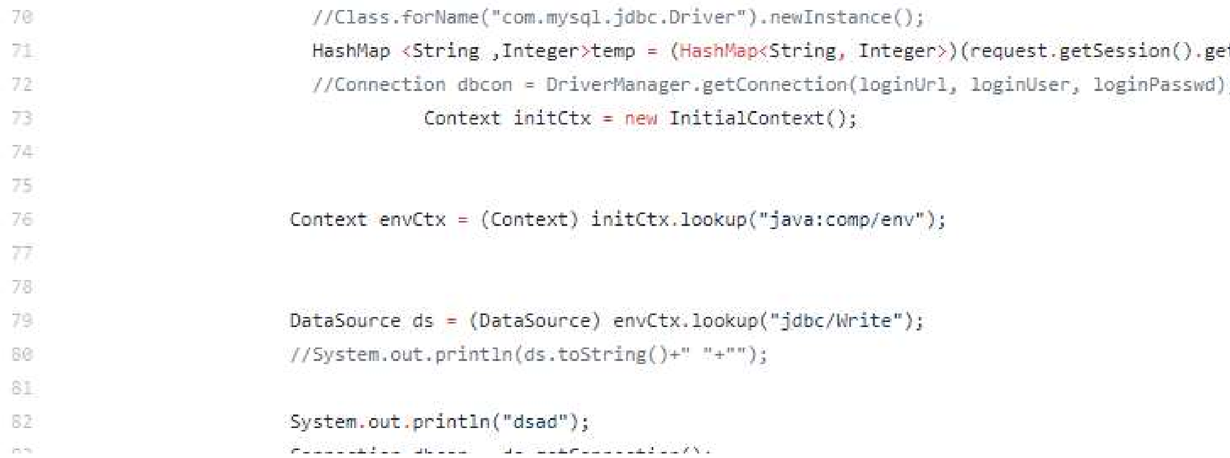
  In dashboard.java line 83

  In checkout.java line 79

  Since dashboard and checkout will write to mysql server, when going to these servlets it will   always go to master's ip.

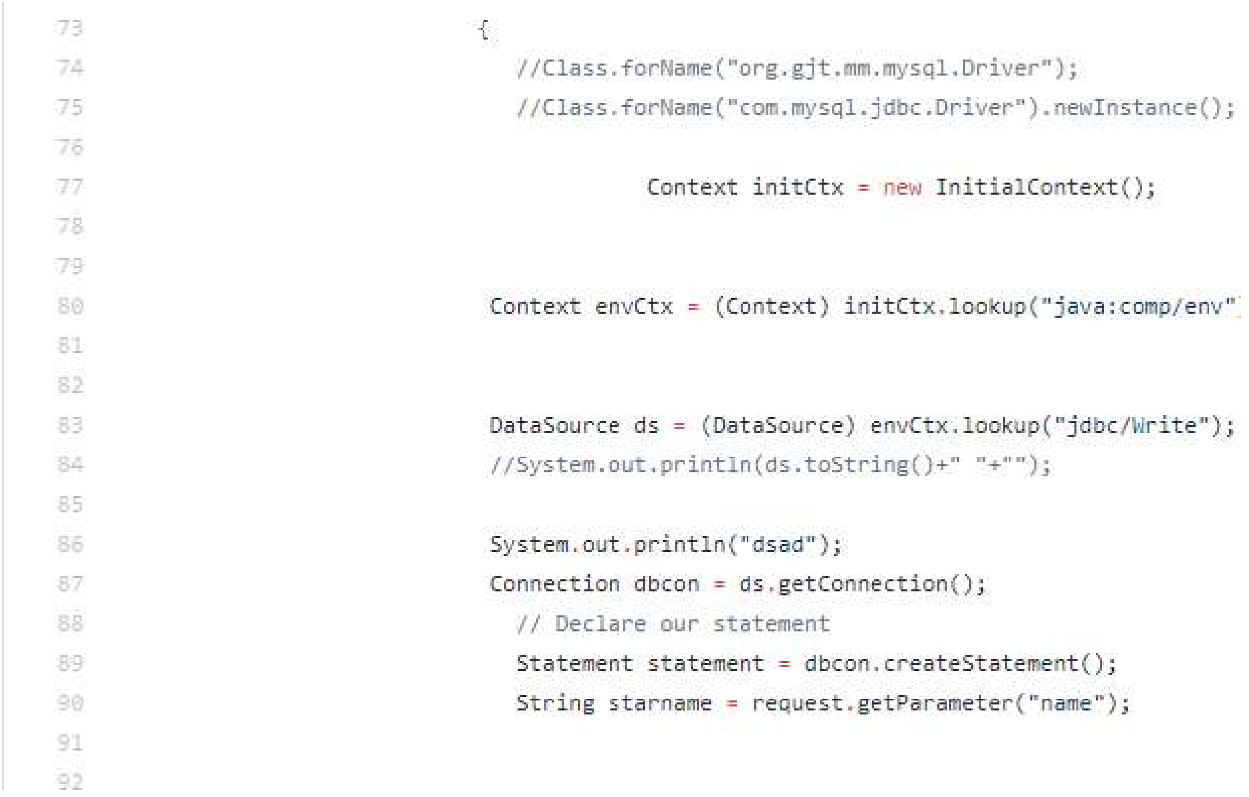
- Snapshots

- I named master mysql to write to they all going to those resources in context.
- Checkout

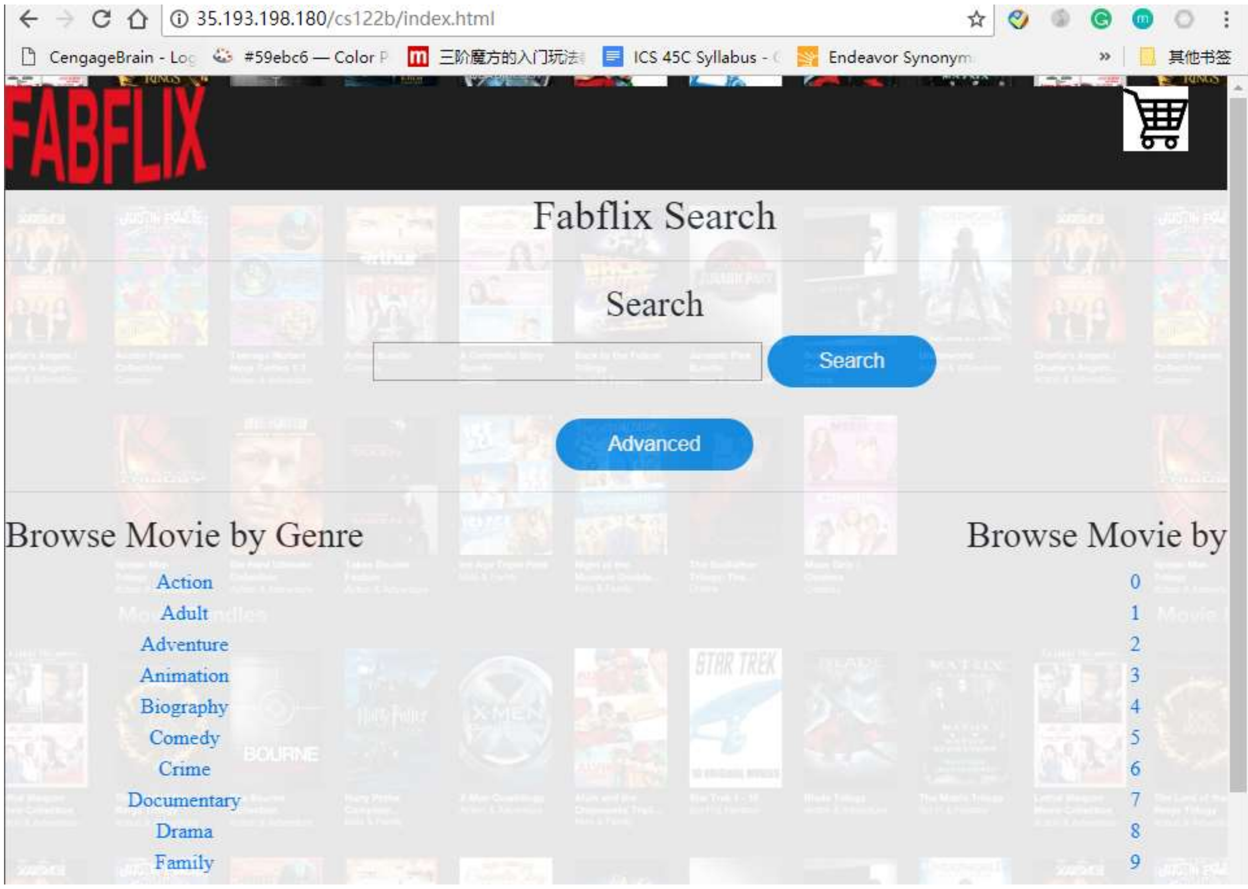


```
70                      //Class.forName("com.mysql.jdbc.Driver").newInstance();
71                      HashMap <String ,Integer>temp = (HashMap<String, Integer>)(request.getSession().ge
72                      //Connection dbcon = DriverManager.getConnection(loginUrl, loginUser, loginPasswd)
73                                Context initCtx = new InitialContext();
74
75
76                    Context envCtx = (Context) initCtx.lookup("java:comp/env");
77
78
79                    DataSource ds = (DataSource) envCtx.lookup("jdbc/Write");
80                    //System.out.println(ds.toString()+" "+"");
81
82                    System.out.println("dsad");
```

- Dashboard



```
        {
          //Class.forName("org.gjt.mm.mysql.Driver");
          //Class.forName("com.mysql.jdbc.Driver").newInstance();

                    Context initCtx = new InitialContext();


         Context envCtx = (Context) initCtx.lookup("java:comp/env"


         DataSource ds = (DataSource) envCtx.lookup("jdbc/Write");
         //System.out.println(ds.toString()+" "+"");

         System.out.println("dsad");
         Connection dbcon = ds.getConnection();
           // Declare our statement
           Statement statement = dbcon.createStatement();
           String starname = request.getParameter("name");


```



35.193.198.180/cs122b/index.html

CengageBrain - Log   #59ebc6 — Color P   三阶魔方的入门玩法   ICS 45C Syllabus -   Endeavor Synonym   »   其他书签

Fabflix Search

Search

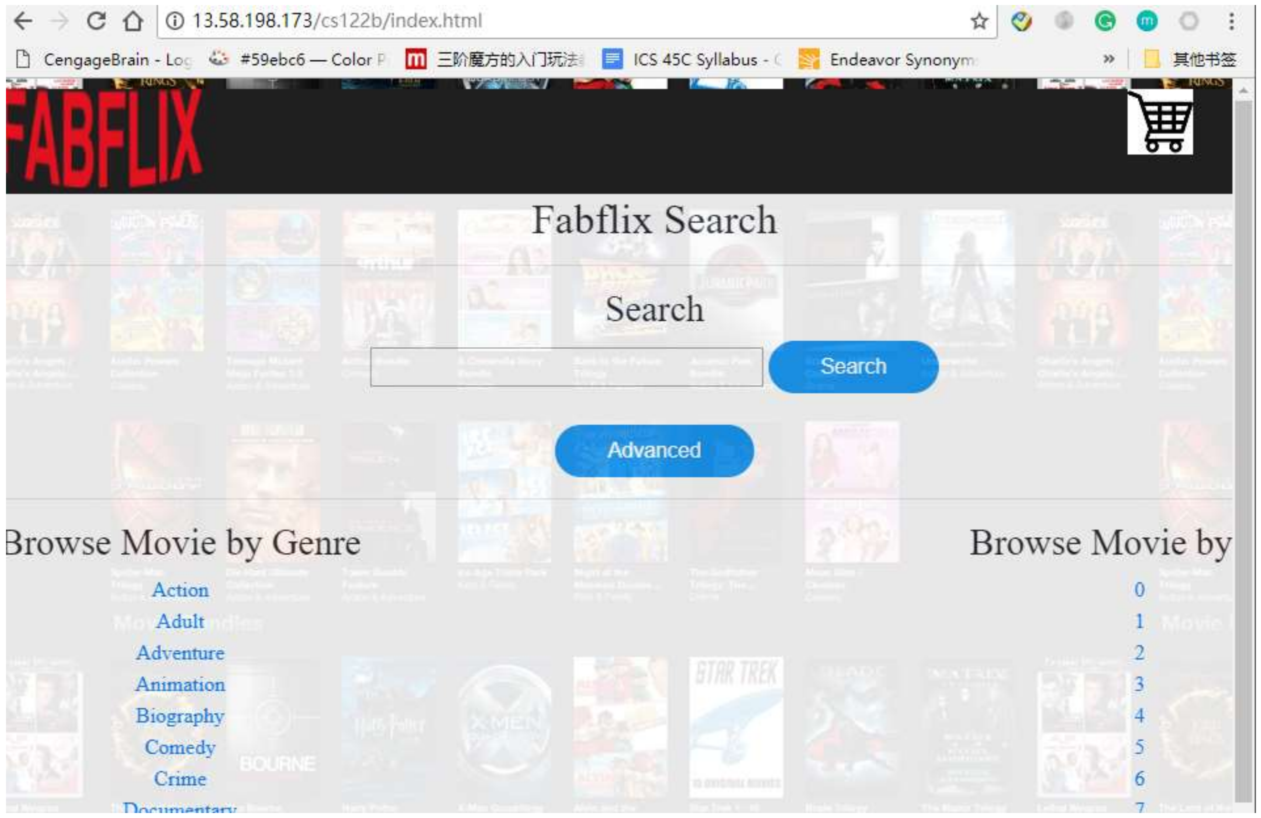
Browse Movie by Genre

Action
Adult
Adventure
Animation
Biography
Comedy
Crime
Documentary
Drama
Family

Browse Movie by

0
1
2
3
4
5
6
7
8
9

-



First one is my google load balancer, and second one is my aws load balancer.

Task 3

- Have you uploaded the log file to Github? Where is it located?
  It is inside a folder called time, which has the TS and TJ of different cases.

- Have you uploaded the HTML file to Github? Where is it located?
- It is also in the time folder called jmeter report1

- Have you uploaded the script to Github? Where is it located?

  It is also in time folder, which is written in python. It will read 2 txt files and print out the average time.

- Have you uploaded the WAR file and README to Github? Where is it located?
  It is directly in the github repository.