CS 122B Group 42 Connection Pooling Reports

Since the method of connection pooling tries to optimize the performance of the program by storing the database connection. It minimizes the number of times it needs to establish connection to the database since it is a very resource-expensive process. Fortunately, we have a stand-alone java servlet to initiate the database connection, so implementing connection pooling was only done in that servlet. We replaced the part of the code that initialize the JDBC connection to call the context.xml and lookup the previously cached connection. In the context.xml file, we changed the context path to point to Fabflix and named the resource to be jdbc/MovieDB. Then, we the JDBC URL and port number to our AWS instance and set the values of the database credentials.

After scaling Fabflix in task 2, we added the master and slave Fabflix instances. This works with connection pooling by storing both the connection resources of the master and slave in the context.xml file. In DatabaseHelper.java source file, we check before getting the data source if it is for the master or slave and return one accordingly.