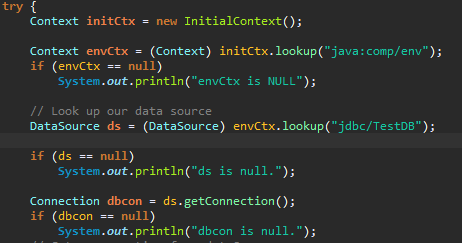
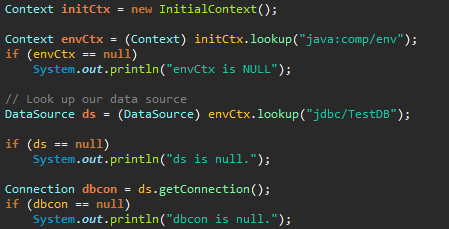
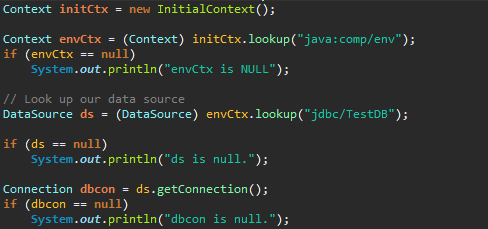
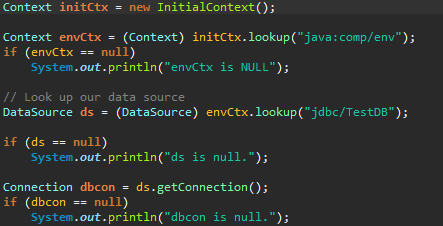
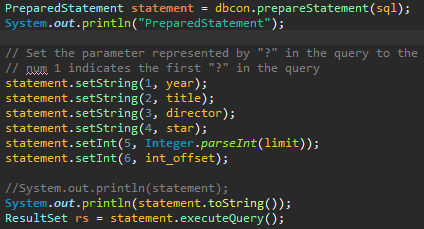
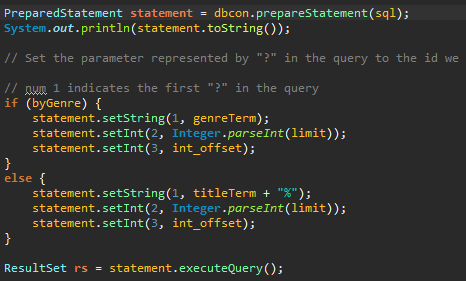
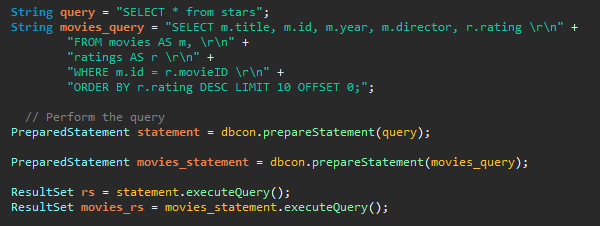
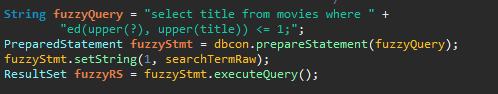
**Task 1**

* How did you use connection pooling?

Whenever a search is used, we include connection pooling. This involves connecting to the database via a Context lookup via the context.xml. Caching prepared statements also helps with connection pooling by saving the query in cache.

* File name, line numbers as in Github
  + AdvancedSearchServlet.java: 216-226
  + BrowseResultServlet.java: 331-345
  + MoviesServlet.java: 68-82
  + SearchServlet.java: 191-205
* Snapshots showing use in your code
  + AdvancedSearchServlet.java
    - 
  + BrowseResultsServlet.java
    - 
  + MoviesServlet.java
    - 
  + SearchServlet.java
    - 
* How did you use Prepared Statements?

Any query that was executed to the database in the search servlets used Prepared Statements. Using any version of set[Type]() allowed us to insert non-SQL information that was is needed to complete the query.

* File name, line numbers as in Github
  + AdvancedSearchServlet.java
    - Lines 235-249
  + BrowseResulstServlet.java
    - Lines 349-365
  + MoviesServlet.java
    - Lines 89-102
  + SearchServlet.java
    - Lines 211-215
* Snapshots showing use in your code
  + AdvancedSearchServlet.java
    - 
  + BrowseResultsServlet.java
    - 
  + MoviesServlet.java
    - 
  + SearchServlet.java
    - 

**Task 2**

* Address of AWS and Google instances

public IPs

------------

Aws slave 18.222.208.184 http://18.222.208.184:8080/fabflix/

Aws master 18.217.148.81 http://18.217.148.81:8080/fabflix/

Aws original 3.17.64.3 http://3.17.64.3/fabflix/

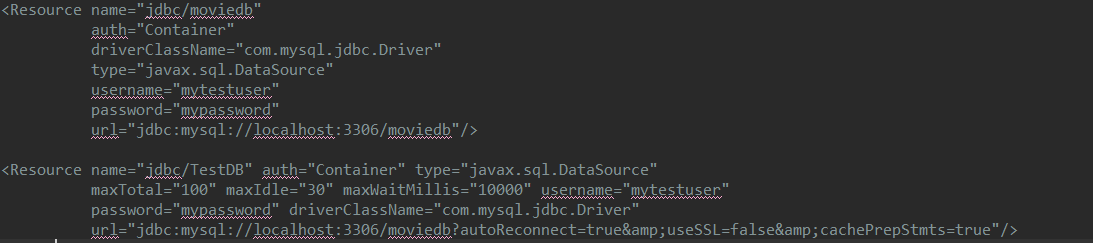
google 35.235.73.254 http://35.235.73.254/fabflix/

* Have you verified that they are accessible? Does Fablix site get opened both on Google’s 80 port and AWS’ 8080 port?

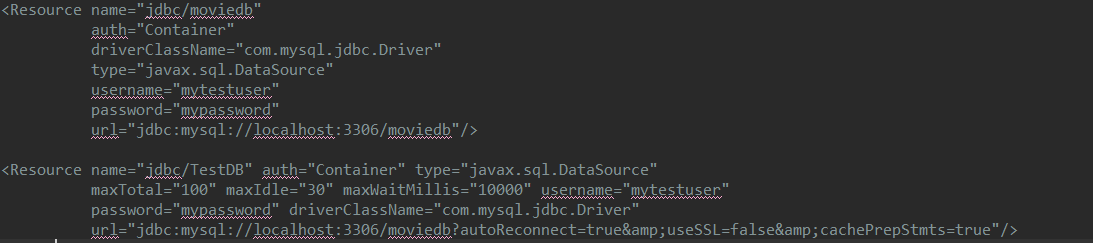
Yes to both questions.

* Explain how connection pooling works with two backend SQL (in your code)?
  + File name, line numbers as in Github

Context.xml: lines 6-17

* Snapshots
* 
* How read/write requests were routed?
  + File name, line numbers as in Github

Context.xml: lines 6-17

* Snapshots
* 

**Task 3**

* Have you uploaded the log files to Github? Where is it located?

Yes, from root: jmeter-report/logs/

Log files in the logs folder:

http-scaled-1-thread

http-scaled-10-threads

http-scaled-10-threads-no-cp

http-scaled-10-threads-no-ps

http-single-1-thread

http-single-10-threads

http-single-10-threads-no-cp

http-single-10-threads-no-ps

https-single-10-threads

There are also copies of the slave/master log files from which the scaled logs were derived. Each log file contains two columns in the format: TJ TQ\n

* Have you uploaded the HTML file (with all sections including analysis, written up) to Github? Where is it located?

Yes, from root: jmeter-report/jmeter\_report.html

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

Yes, it is located at the root as well as in jmeter-report/average.py

* Have you uploaded the WAR file and README to Github? Where is it located?

Yes, both are at the root.