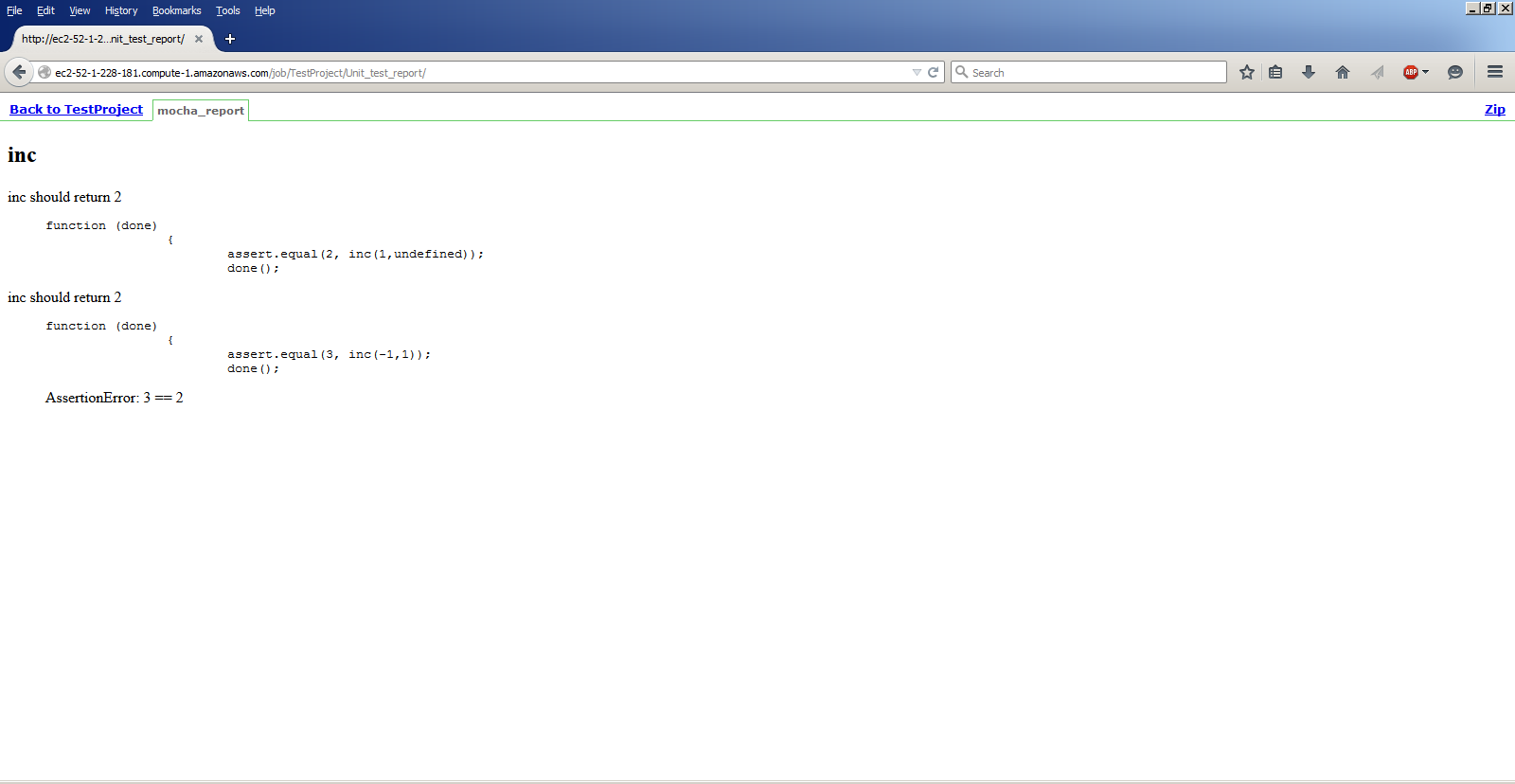
Task 1:

The ability to run unit tests, measure coverage, and report the results

Mocha has been used for unit testing the JavaScript program and generate a HTML report. Jenkins HTML publisher plugin is used to produce the unit test report

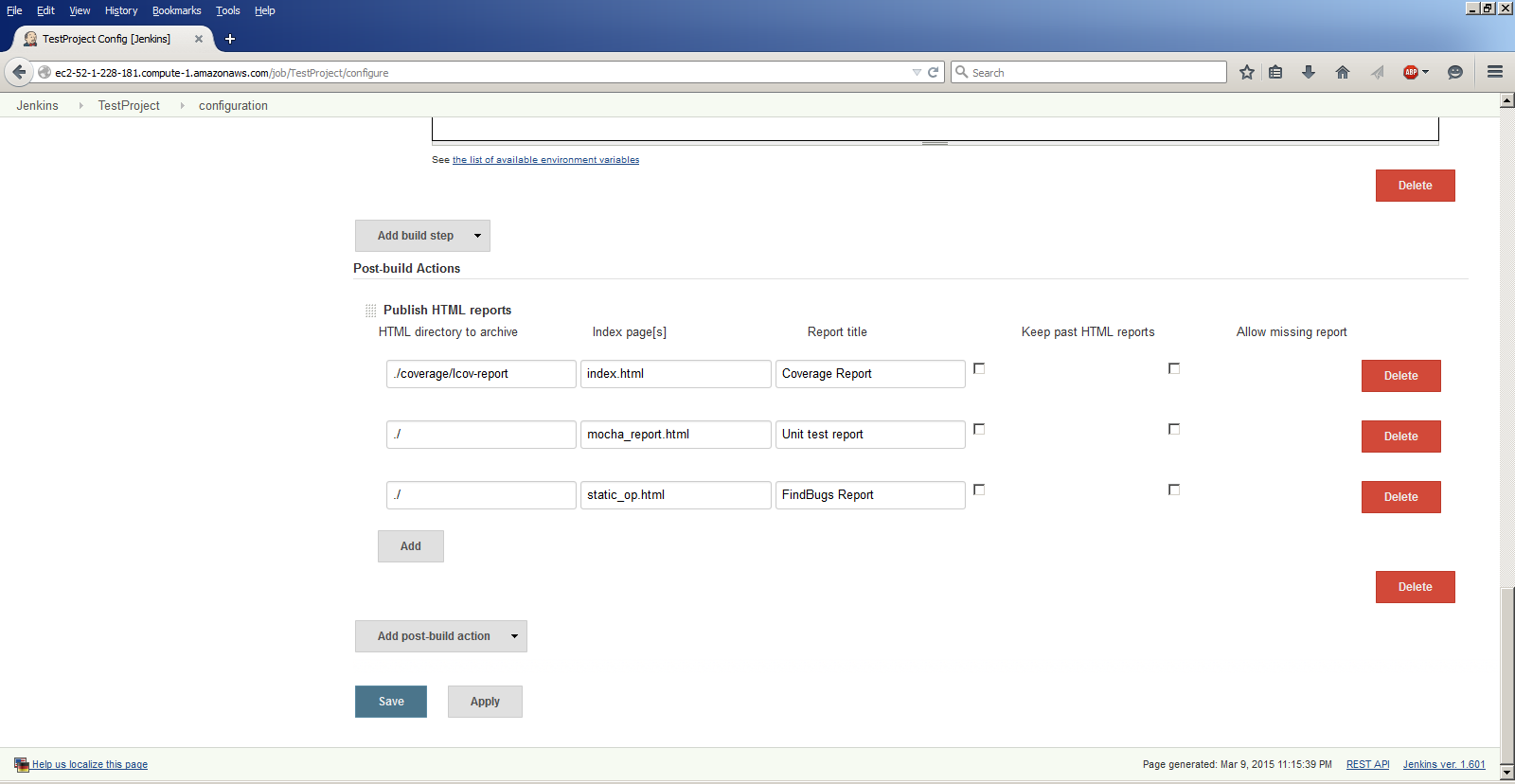


Task 2:

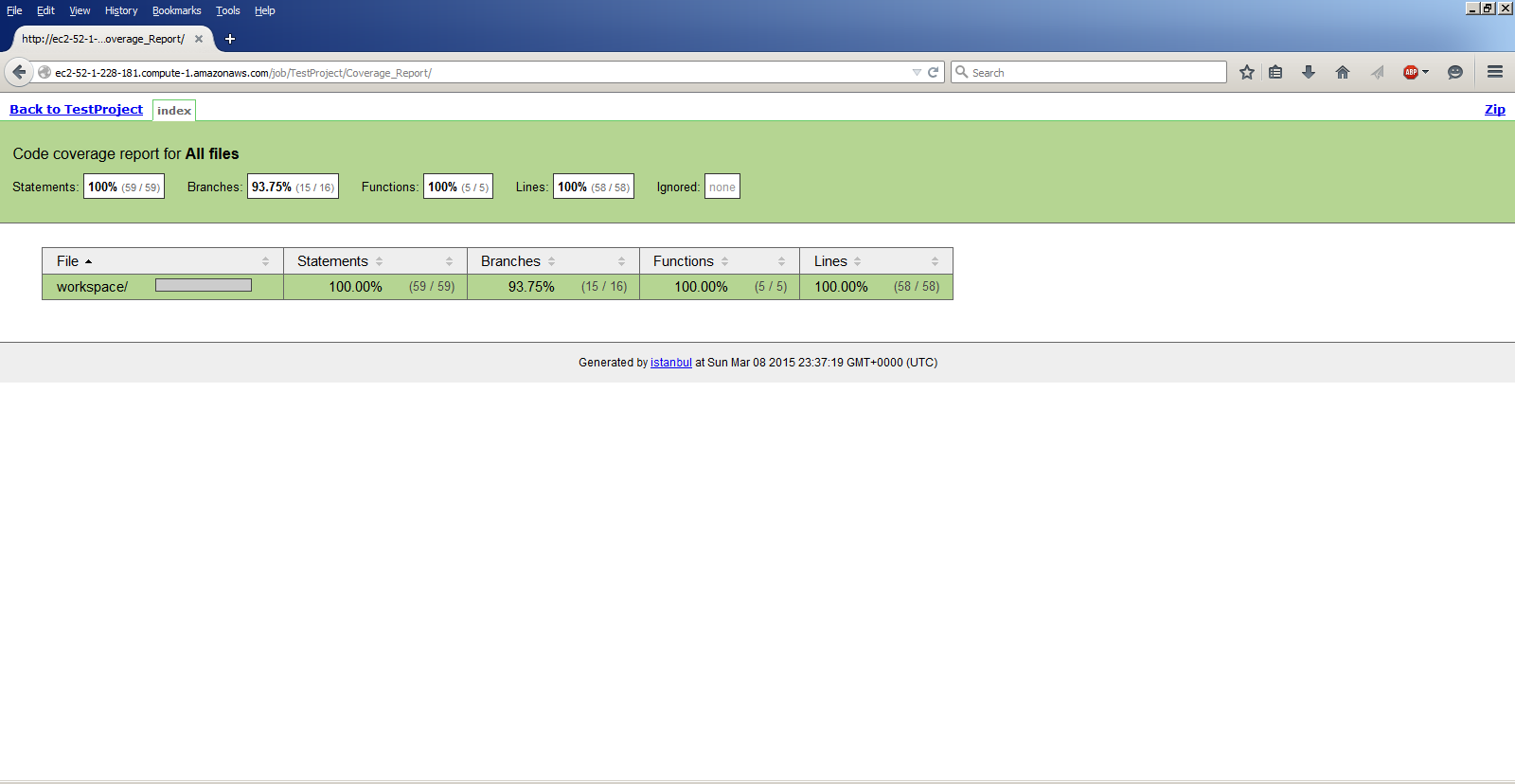
The ability to improve testing coverage using one of the techniques covered in class: constraint-based test generation, fuzzing, etc. You can use an existing tool or implement your own approach.

Constraint-based testing has been used to improve test coverage and a HTML report is produced for Jenkins use

HTML files path spec for unit test, static analysis and coverage:



Unit test report:



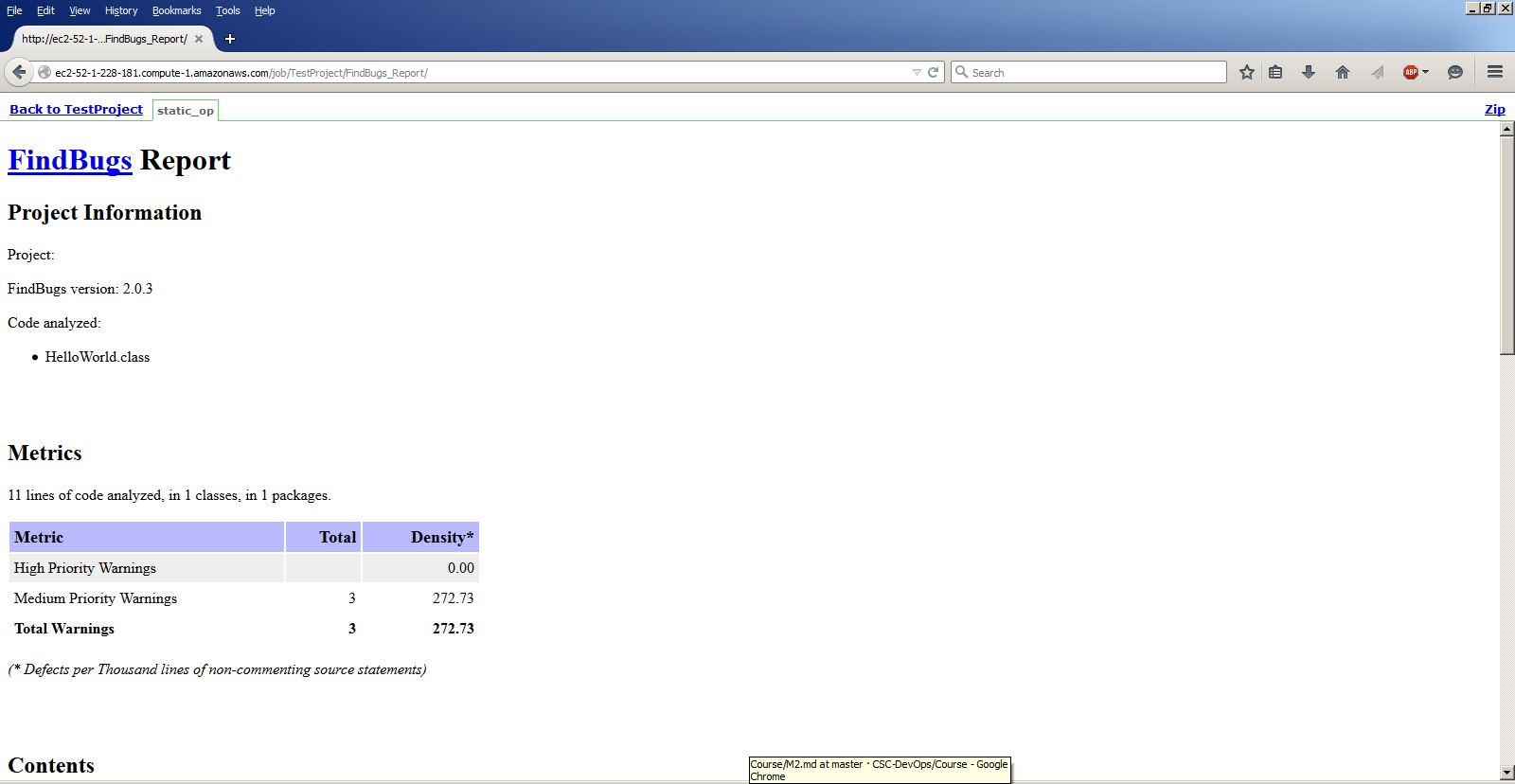
Task 3:

The ability to run an existing static analysis tool on the source code (e.g. FindBugs, PMD, CheckStyle, NCover, Lint, etc.), process its results, and report its findings

FindBugs has been used to run static analysis on a Java program in the repo. A sample Java program that uses + for string concatenation has been used to show the FindBugs warning:

SBSC: Method concatenates strings using + in a loop(SBSC\_USE\_STRINGBUFFER\_CONCATENATION)

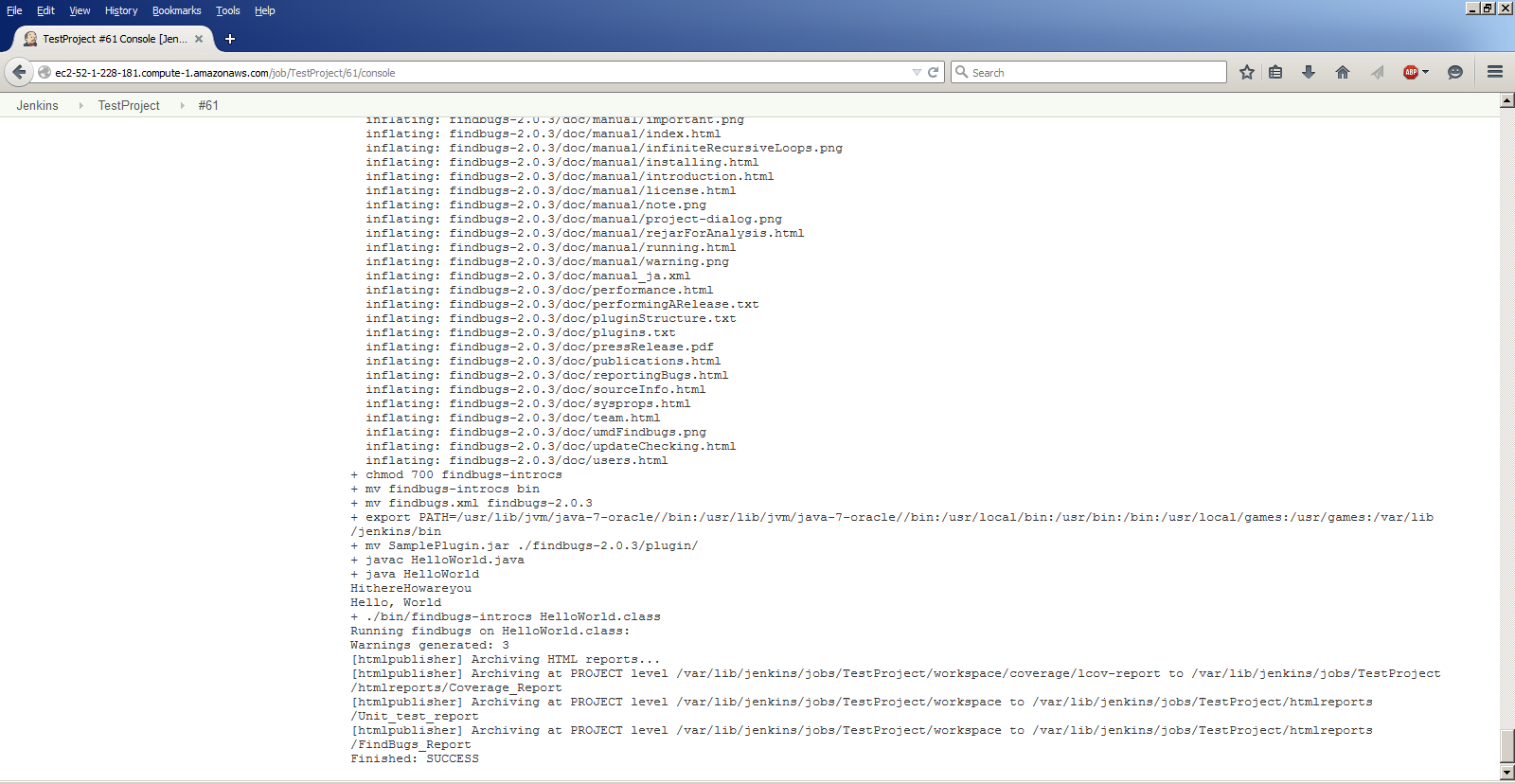
For reporting the findings, a HTML file is generated using the findbugs command line option and is used in Jenkins.



Task 4:

The ability to extend an existing analysis tool with a custom rule, or implement a new analysis. For example, you could write a static analysis that checks for the ratio of comments to code, or finds parse errors in SQL string statements. You could introduce security checks, a dynamic analysis, a data-flow analysis or a data-flow based test coverage

A custom FindBugs detector plugin that alerts usage of System.out statements in the Java program has been used to showcase this ability.

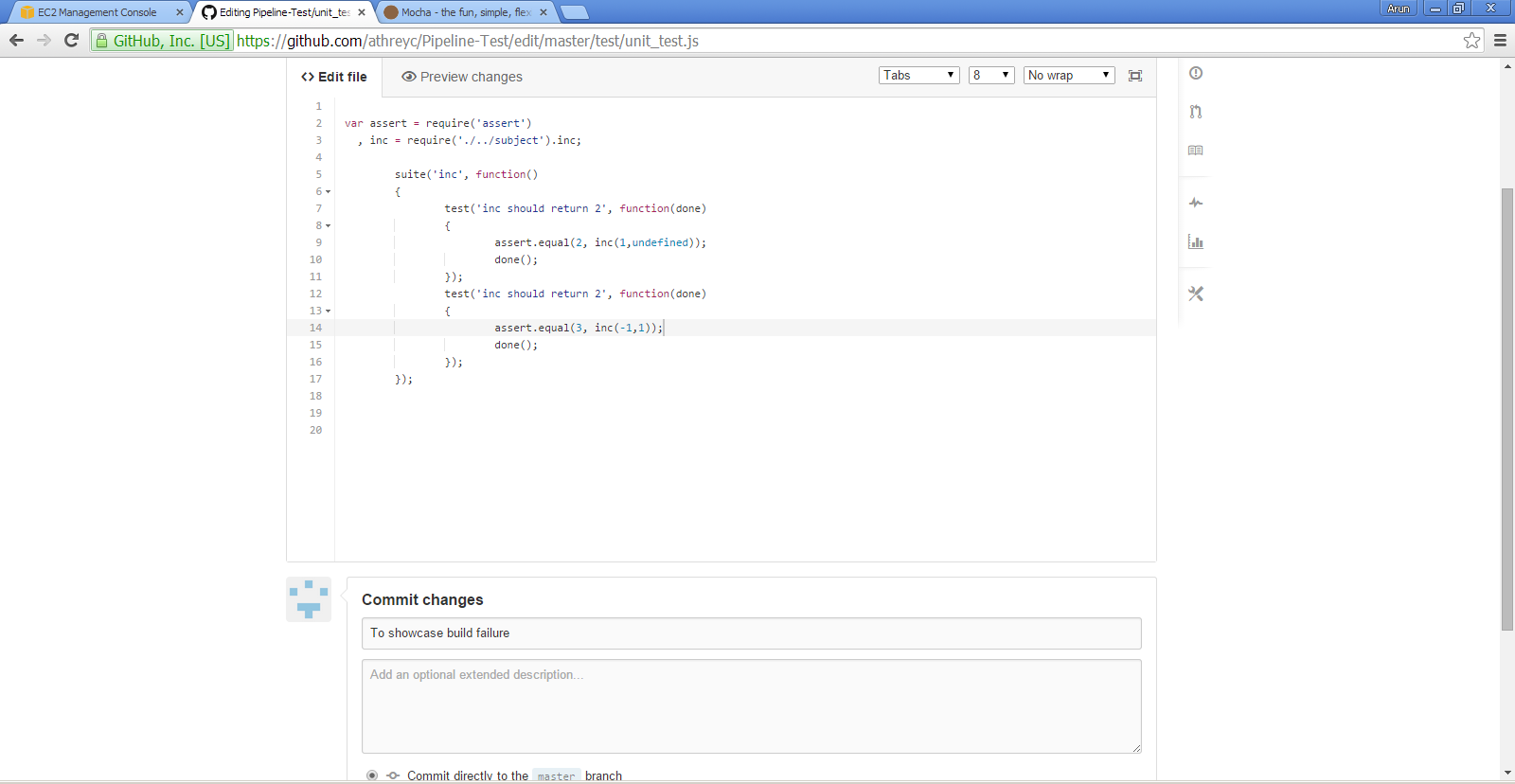


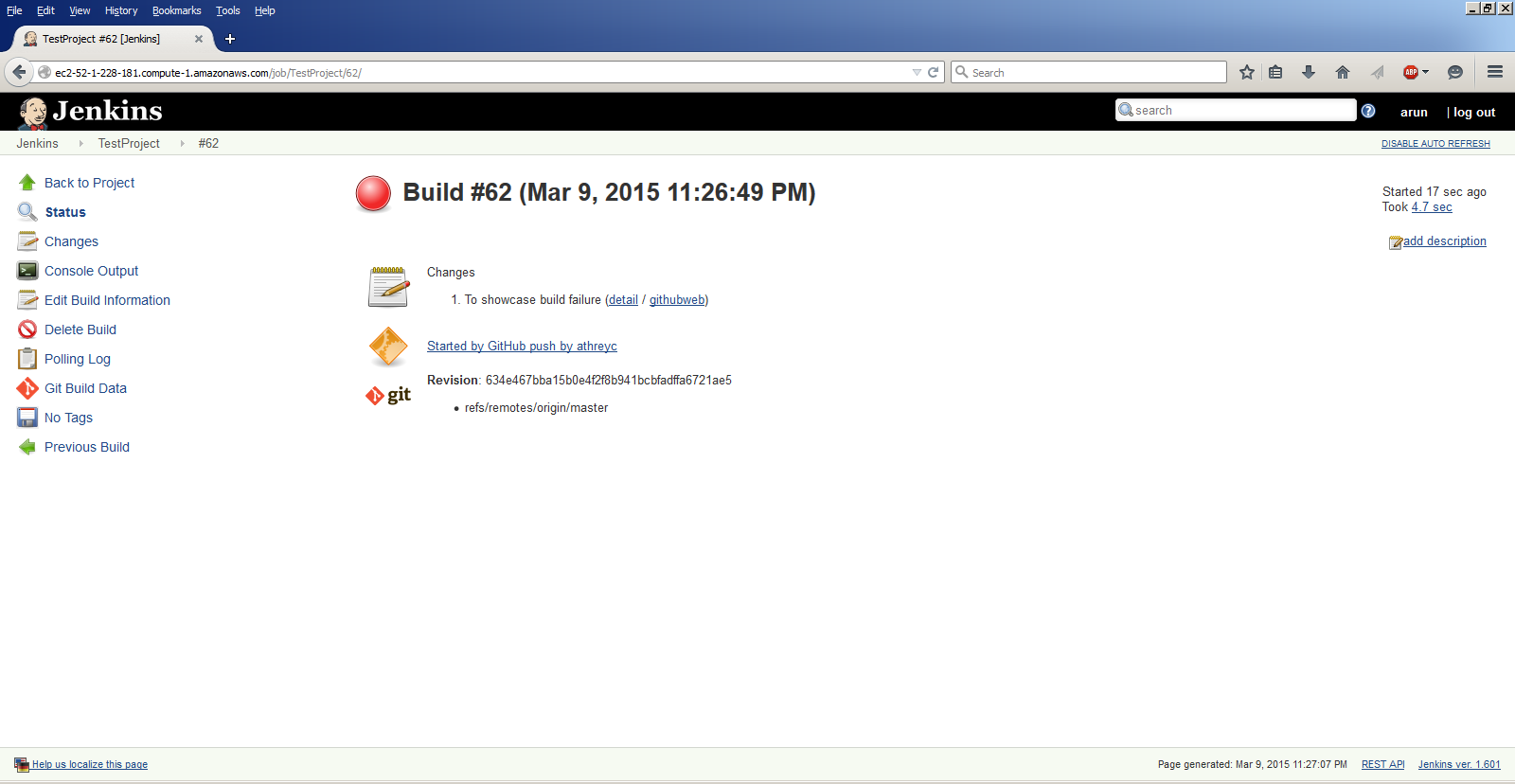
Task 5:

The ability to reject a commit if it fails a minimum testing criteria (e.g. failed test case, or less than 50% statement coverage) and analysis criteria (e.g. cannot commits that generate a particular FindBugs rule, such as "Method concatenates strings using + in a loop").

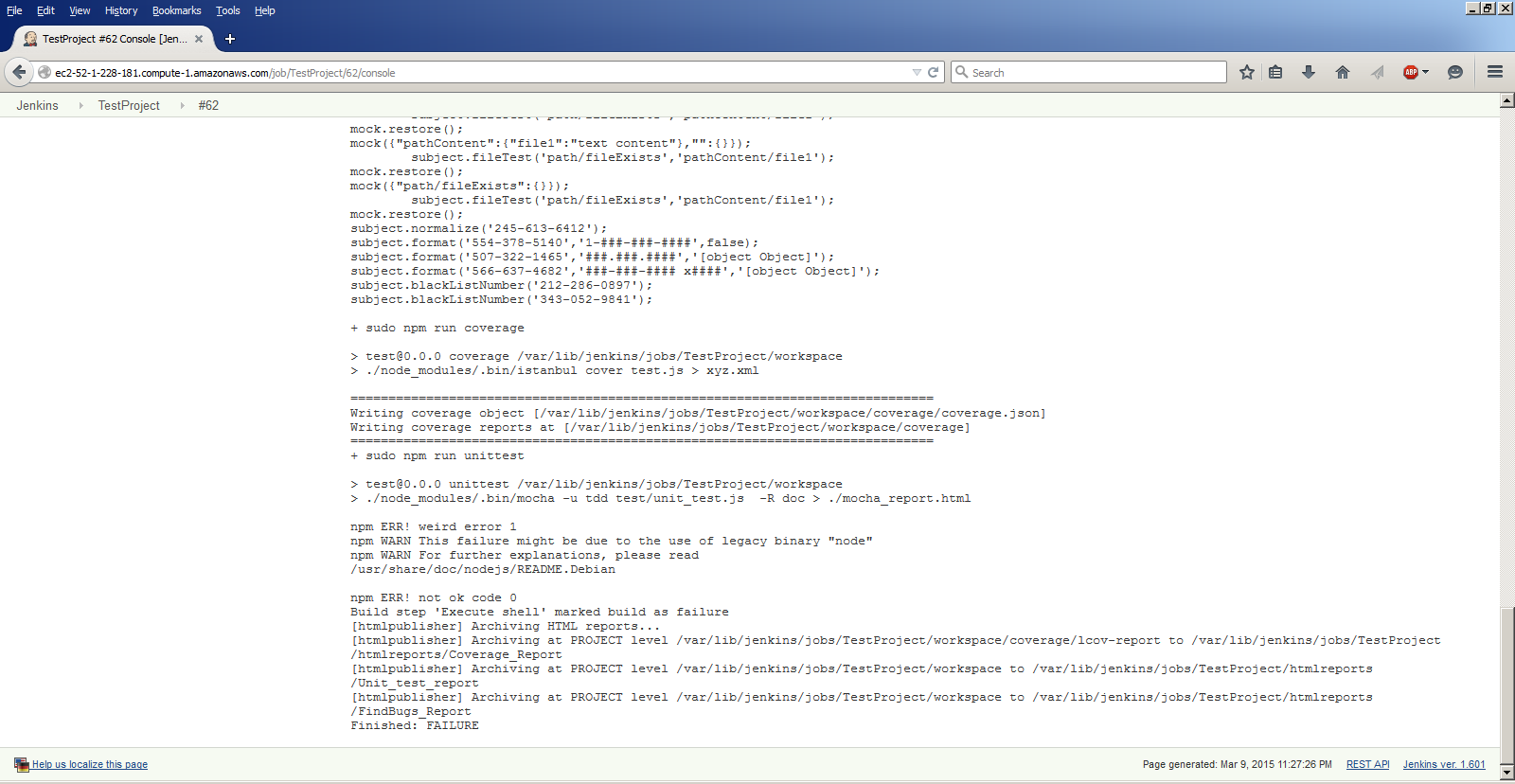
Unit test failure:

For failing a unit testing, a Mocha test is written to effect a failure of one of the tests. This results in the return of a non-zero value for one of the pre-build shell scripts that prompts a failure of the build.





Build failure:



Analysis failure:

