Kyua and Jenkins
Testing Framework for BSD

by Craig Rodrigues
rodrigc@FreeBSD.org

Kyua

What is Kyua?

- It's a pun on “Q.A.”, but pronounced “kyoo-ah”
- It provides a framework for defining tests, and running them
- Actual tests can be implemented in C, C++, Shell, Python, etc.

History of Kyua

- Automated Testing Framework (ATF) started as a NetBSD Google Summer of Code project in 2007 by Julio Merino (jmmv@)
- ATF was targeted to testing NetBSD with a consistent framework
- Imported into NetBSD, and later FreeBSD

History of Kyua

- ATF was later refactored into two parts
  - ATF: libraries and API's for writing tests in C, C++, shell
  - Kyua: run-time engine for running the tests, and generating reports

Who uses Kyua?

- Open Source projects:
  - NetBSD: ATF has been imported into base system
  - FreeBSD: ATF has been imported into base system
- Companies building BSD based products

Trying out Kyua under FreeBSD

Build tests included with FreeBSD

- In FreeBSD-10 and lower, put in /etc/make.conf **WITH_TESTS=”yes”**
- make buildworld; make installworld
- Tests will be in /usr/tests

Install kyua

- pkg install devel/kyua

List the tests

- cd /usr/tests
- kyua list

Run the tests

- cd /usr/tests
- kyua test

Generate reports

- Plain text report
  - kyua report
- HTML report
  - kyua report-html
- Junit XML report
  - kyua report-junit

Writing your own tests

What do you need?

- Kyuafile
  - kyua binary reads this to figure out what tests to run
- Tests, can be one or more of:
  - plain test
  - ATF test
  - TAP test

Kyuafile

```
-- Comments in Kyuafiles must start with
-- two hyphens

-- Need this boilerplate:

syntax(2)


-- The name of the test suite must be
-- defined.

test_suite('mytests')
```

Kyuafile (continued)

```
-- Specify test programs, must be in same
dir
-- as Kyuafile

plain_test_program{name='test1.sh'}

plain_test_program{name='test2.sh'}

atf_test_program{name='test3.sh'}

tap_test_program{name="legacy_test"}


 -- Include another Kyuafile, to get more
 -- tests in different dir

 include(“otherdir/Kyuafile”)
```

Plain test program

- exit status 0, means test succeeded
- exit status non-zero means test failed
- One test case per program

ATF test program

- Must use the ATF libraries
- Available only for C, C++, Bourne shell programs
- API allows you to specify pass, fail, skipped
- Single program can contain multiple test cases

ATF shell example

```
#!/usr/local/bin/atf-sh

atf_test_case test1
test1_head()
{
    atf_set "descr" "Test 1, remove file"
}
```

ATF shell example (continued)

```
test1_body()
{
    rm /something
}
atf_init_test_cases()
{
    atf_add_test_case test1
}
```

ATF shell hints

- Always do “kyua list” to make sure test cases are properly integrated
- Never explicitly exit or return in your test cases. This will mess things up.
- Look at libatf-sh.subr to see the shell routines used for executing testss

TAP test program

- Program generates output which follows the “Test Anything Protocol” http://testanything.org
- “kyua list” does not work for TAP, need to run the tests to figure out how many there are

TAP test sample output

```
1..4
ok 1 - Input file opened
not ok 2 - First line of the input valid
ok 3 - Read the rest of the file
not ok 4 – Other failure
```

Jenkins

Jenkins and Continuous Integration

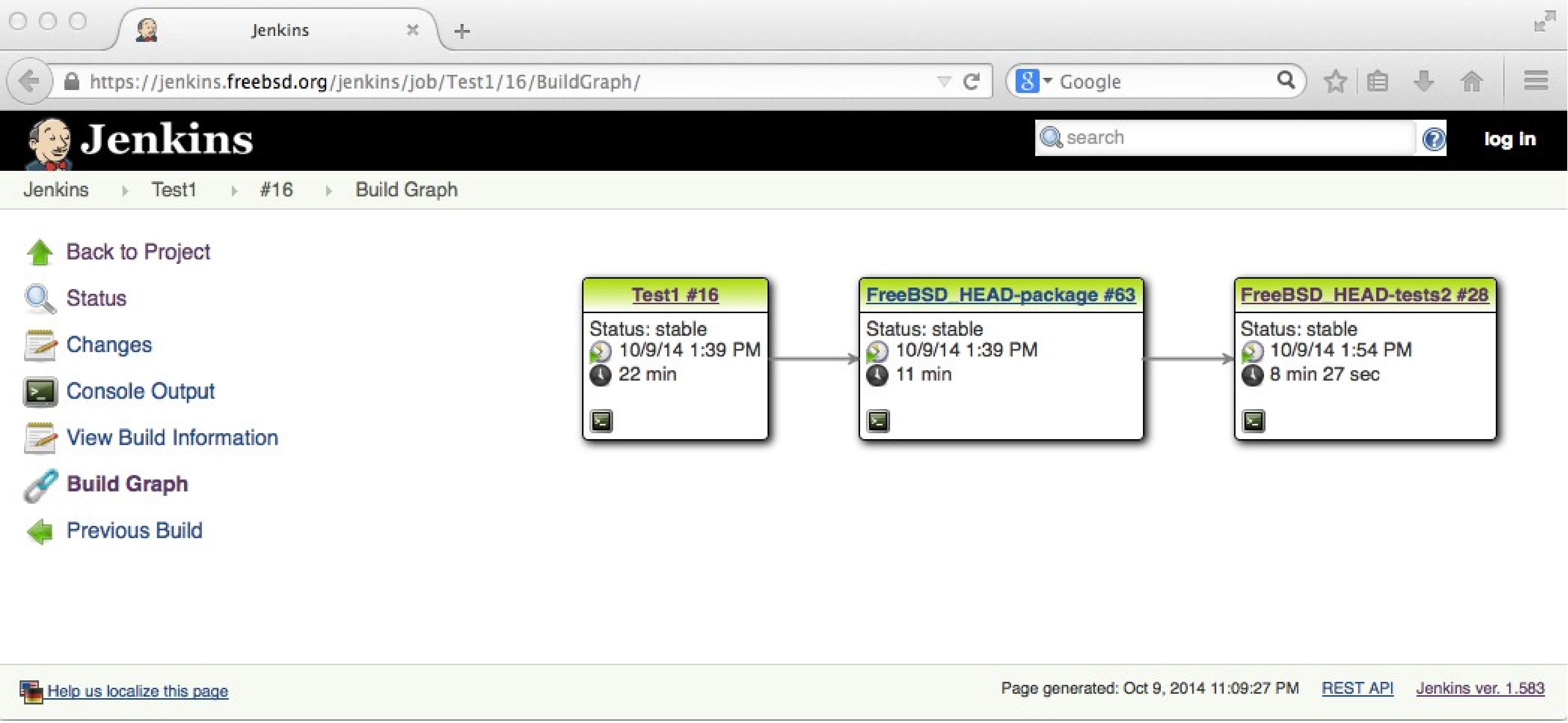
- Jenkins can be used for Continuous Integration
  - monitor SCM (git, svn) to see new checkins
  - build the code
  - run tests
  - produce test reports

Jenkins and Testing

- Jenkins does not have a native test execution engine, but can run external ones like JUnit
- Jenkins can take JUnit XML formatted test result output and import it natively.

Jenkins and Kyua

- To integrate Jenkins and Kyua, create a Jenkins job which runs these commands:
  - kyua test
  - kyua report-junit
- Configure Jenkins job to **Publish JUnit test result report**

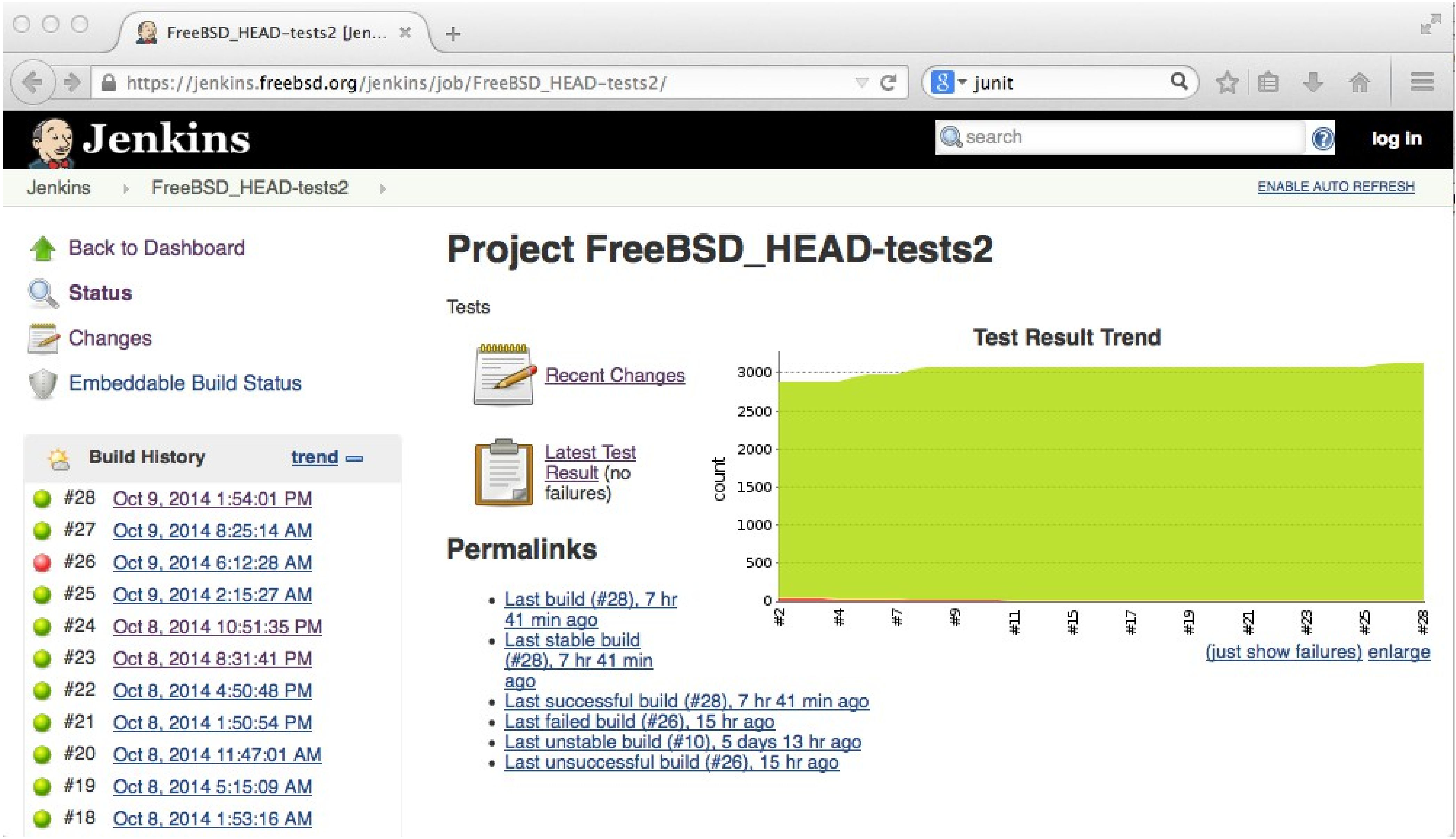


Test1 #16
Status: stable
10/9/14 1:39 PM
22 min

FreeBSD_HEAD-package #63
Status: stable
10/9/14 1:39 PM
11 min

FreeBSD_HEAD-tests2 #28
Status: stable
10/9/14 1:54 PM
8 min 27 sec

Page generated: Oct 9, 2014 11:09:27 PM



Project FreeBSD_HEAD-tests2

Tests

Recent Changes

Latest Test Result (no failures)

Build History trend

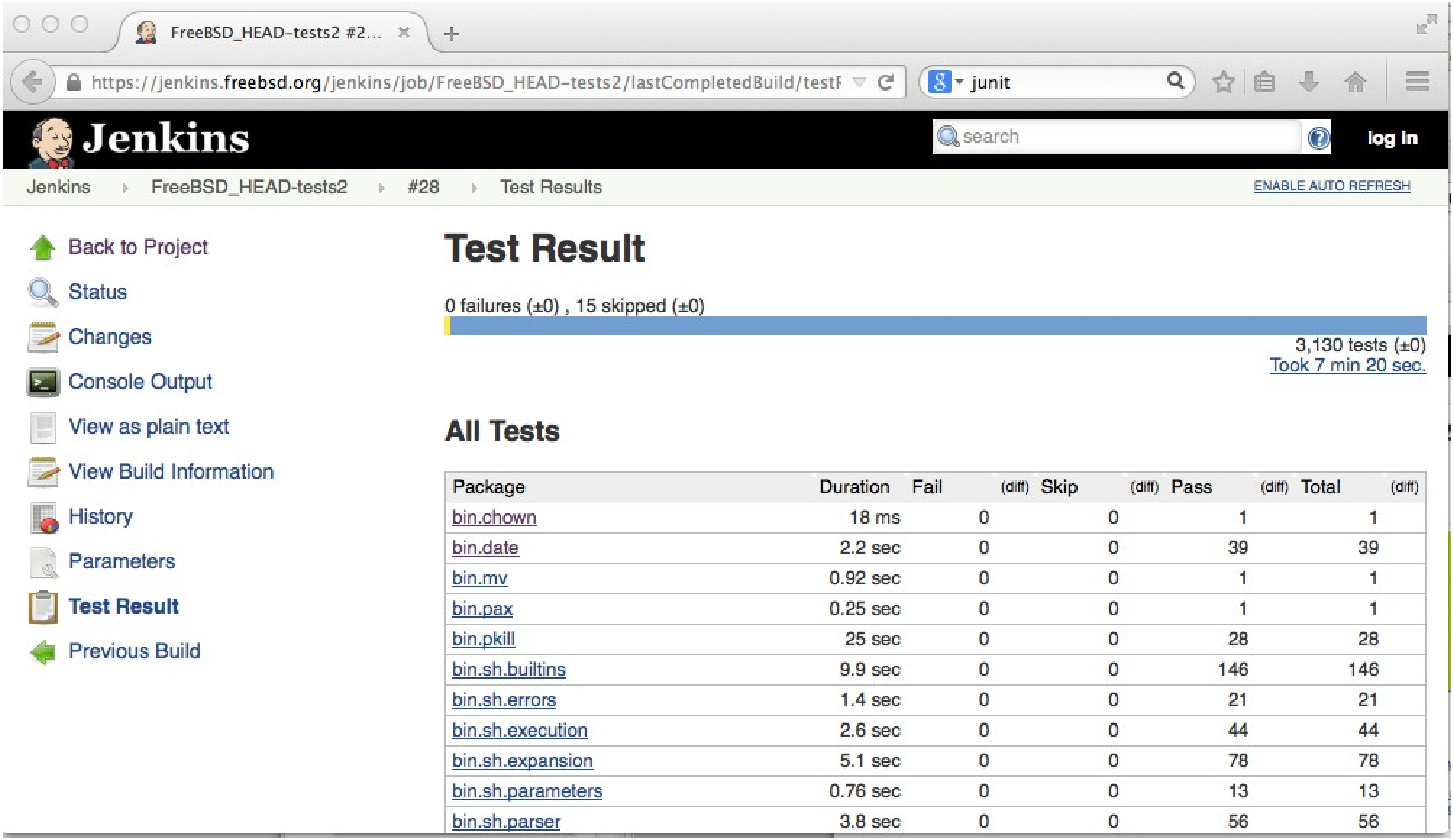
🟢	#28	Oct 9, 2014 1:54:01 PM
🟢	#27	Oct 9, 2014 8:25:14 AM
🔴	#26	Oct 9, 2014 6:12:28 AM
🟢	#25	Oct 9, 2014 2:15:27 AM
🟢	#24	Oct 8, 2014 10:51:35 PM
🟢	#23	Oct 8, 2014 8:31:41 PM
🟢	#22	Oct 8, 2014 4:50:48 PM
🟢	#21	Oct 8, 2014 1:50:54 PM
🟢	#20	Oct 8, 2014 11:47:01 AM
🟢	#19	Oct 8, 2014 5:15:09 AM
🟢	#18	Oct 8, 2014 1:53:16 AM

Test Result Trend

(just show failures) enlarge

Permalinks

- Last build (#28), 7 hr 41 min ago
- Last stable build (#28), 7 hr 41 min ago
- Last successful build (#28), 7 hr 41 min ago
- Last failed build (#26), 15 hr ago
- Last unstable build (#10), 5 days 13 hr ago
- Last unsuccessful build (#26), 15 hr ago



Test Result

0 failures (±0) , 15 skipped (±0)

3,130 tests (±0)
Took 7 min 20 sec.

All Tests

Package	Duration	Fail	(diff)	Skip	(diff)	Pass	(diff)	Total	(diff)
bin.chown	18 ms	0		0		1		1	
bin.date	2.2 sec	0		0		39		39	
bin.mv	0.92 sec	0		0		1		1	
bin.pax	0.25 sec	0		0		1		1	
bin.pkill	25 sec	0		0		28		28	
bin.sh.builtins	9.9 sec	0		0		146		146	
bin.sh.errors	1.4 sec	0		0		21		21	
bin.sh.execution	2.6 sec	0		0		44		44	
bin.sh.expansion	5.1 sec	0		0		78		78	
bin.sh.parameters	0.76 sec	0		0		13		13	
bin.sh.parser	3.8 sec	0		0		56		56	

Further references

- Links:
  - http://wiki.freebsd.org/Jenkins
    http://github.com/jmmv/kyua
  - https://github.com/rodrigc/kyua/wiki/Quickstart-Guide
  - http://jenkins-ci.org/content/freebsd-project-use-jenkins-os-testing
- Mailing lists:
  - freebsd-testing@FreeBSD.org
  - kyua-discuss@googlegroups.com

Needs/wants and next steps

- More people writing unit tests, looking at test results, fixing things
- Need other tests: ZFS, network, performance, etc.
- More builds/tests on other architectures
- Better devops for spinning up new build environments (Saltstack, Chef, Puppet, ??)