Continuous Integration

Ádám Révész, Attila Ulbert, Norbert Pataki, Zoltán Gera

Eötvös Loránd University,
Budapest, Hungary

Introduction to
Cloud Technologies

Contents

Introduction

- Software development – teamwork
- Teamwork needs tools
  - Version control: svn, git, Perforce, etc.
  - Command-line build tools: CMake, gradle, maven, etc.
  - Bug tracking/Issue tracking tools: Bugzilla, Atlassian Jira, etc.

Motivating Problems

- The version control cannot guarantee that the code in the repository is “correct”
  - The code may cannot be compiled or essential unit tests fail
  - When the developers can update their local checkout?
  - Which parts of the software can be updated in the local checkout?
  - Which previous revision is proper/stable?
  - Tester: “This feature doesn’t work on my computer.” Developer: This feature does work on my computer pretty well.”
  - Developer: “The tests pass on my computer.”
  - When a feature or a test case went wrong?
  - Who is the responsible for the problem? Who commited into the version control? Who did not commit a referred code?
- Does a bugfix causes new problems?

Motivation

- Frequent releases – at least once per month
- Radically reduce integration costs – avoid integration hell
- Reduce the costs of finding the cause of an abnormal behavior

Definition

- Software development practice
- Frequent integration of the developed code and baseline
- Automated build with QA
  - compilation
  - unit, component, integration, end-to-end tests
  - static analysis, code coverage, etc.
- Transparent reports
- XP development method

Principles

- Maintain a Single Source Repository
- Automate the Build – IDE is not a build system
- Make Your Build Self-Testing
- Everyone Commits To the Mainline Every Day
- Every Commit Should Build the Mainline on an Integration Machine

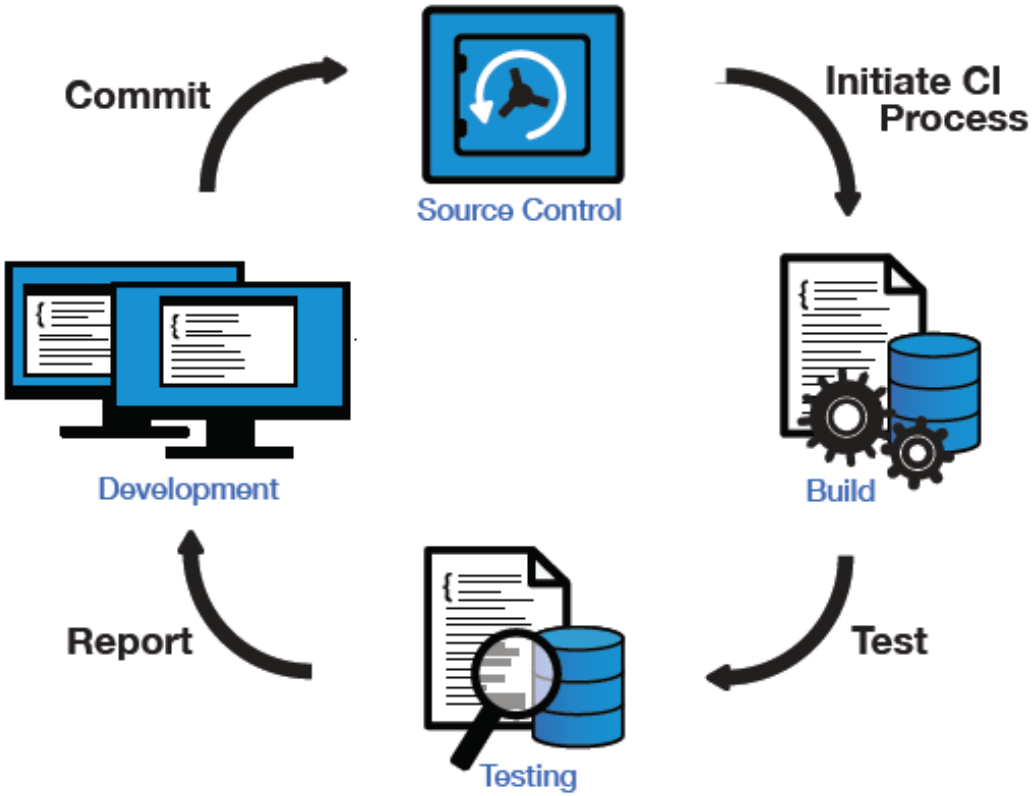
Principles

- Fix Broken Builds Immediately
- Keep the Build Fast (split the tasks, but unfortunately, tests may take longer)
- Test in a Clone of the Production Environment
- Make it Easy for Anyone to Get the Latest Executable
- Everyone can see what's happening
- Automate Deployment

Benefits

- Quickly reveals that the code in the repository does not compile
- Quickly reveals if the code does not pass a testcase
- It may reveal which commit breaks the build
- It can be seen which is the latest working revision
- Previous builds data (console log, test running, etc.) is stored and available
- Report: e-mail, Slack, web ui, etc.
- Referential point - the definitive machine to decide if a test fails

CI Cycle


Commit
Source Control
Initiate CI Process
Development
Build
Report
Testing
Test

Importance of Fast Build Jobs

- Runtime of Build and Test Jobs is important
- Not so many changes in the baseline during execution
- What are changes since the jobs was successful?
- Deterministic test cases
- Parallel, split jobs, Master-slave CI architecture
- Multi-stage build processes

Build Times

- Compilation with Unit Tests: should be under 20 minutes
- Unit testcase: under 1 second.
- Speed-up:
  - Mocked features
  - In-memory Databases
  - Removing dead code
- Slow processes (e.g. security) – nightly build

CI – Lava lamp, Build light indicator

Tools

- Cruise Control (`http://cruisecontrol.sourceforge.net/`)
- Jenkins/Hudson (`https://jenkins-ci.org/`)
- IBM Rational Team Concert
- Microsoft Team Foundation Server
- Travis CI (`https://travis-ci.org/`)
- GoCD (`https://www.gocd.org/`)
- etc.

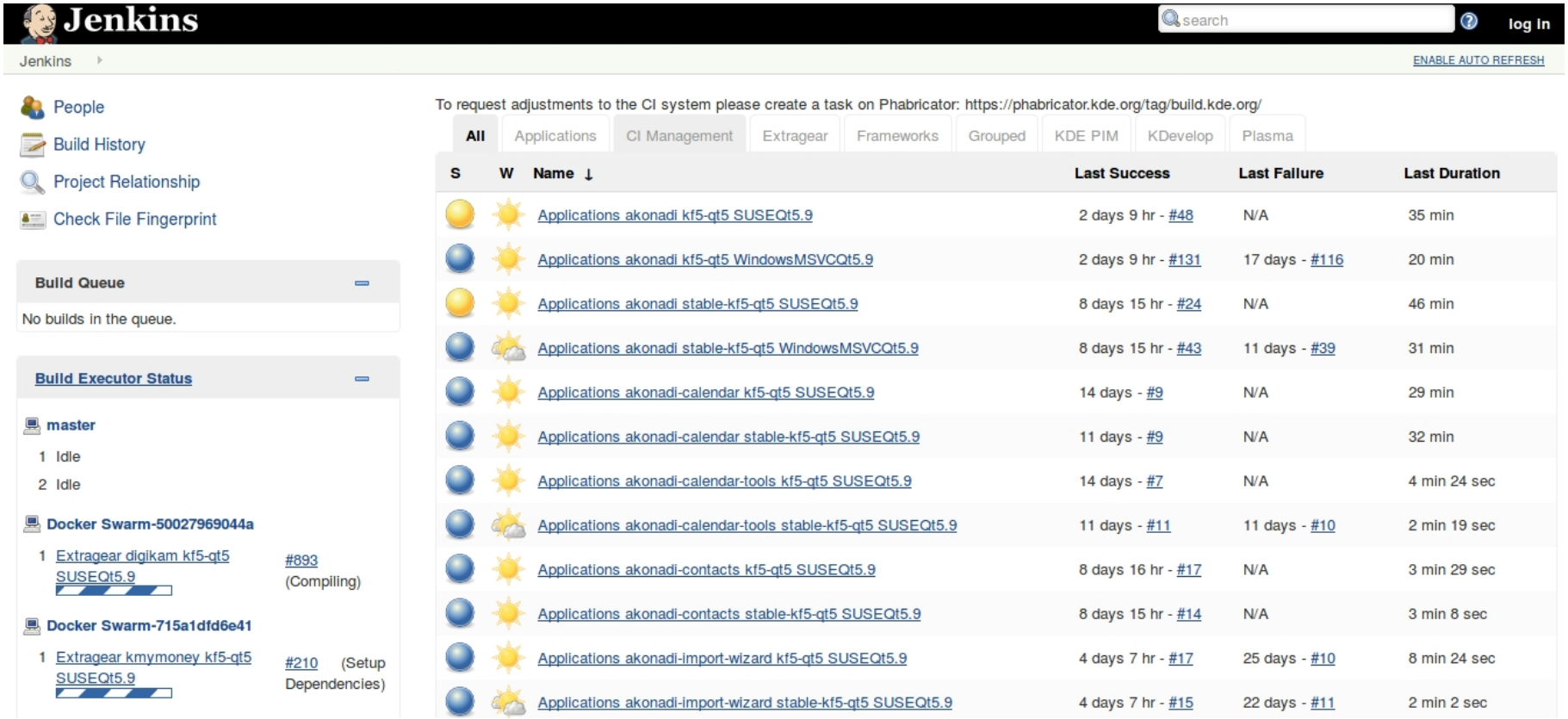
Jenkins



Jenkins

- CI tool, system
- Widely-used
- Open source, multiplatform CI server
- Web UI, REST API
- Various configuration options
- Many useful available plugins
- Pipelines – Jenkins 1 vs. Jenkins 2

The Jenkins Dashboard



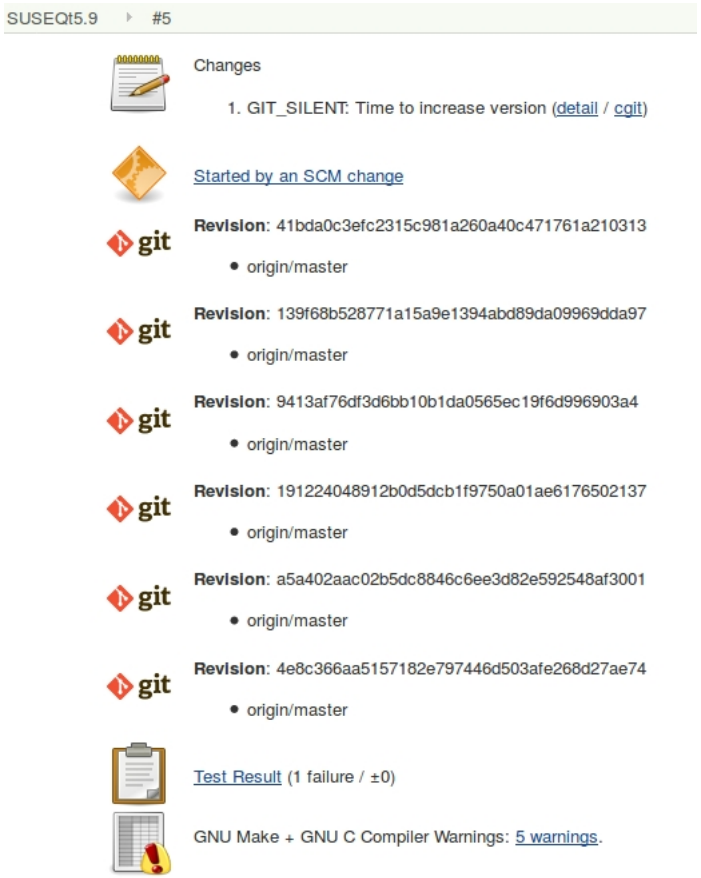
Jobs

Applications kdegraphics-thumbnailers kf5-qt5 FreeBSDQt5.9	17 days - #1	N/A	5 hr 20 min
Applications kdegraphics-thumbnailers kf5-qt5 SUSEQt5.9	1 mo 15 days - #1	N/A	18 min
Applications kdegraphics-thumbnailers stable-kf5-qt5 FreeBSDQt5.9	17 days - #1	N/A	9 hr 12 min
Applications kdegraphics-thumbnailers stable-kf5-qt5 SUSEQt5.9	1 mo 12 days - #2	N/A	1 hr 1 min
Applications kdenetwork-filesharing kf5-qt5 FreeBSDQt5.9	17 days - #1	N/A	5 hr 3 min
Applications kdenetwork-filesharing kf5-qt5 SUSEQt5.9	1 mo 15 days - #1	N/A	32 min
Applications kdenetwork-filesharing stable-kf5-qt5 FreeBSDQt5.9	17 days - #1	N/A	8 hr 10 min
Applications kdenetwork-filesharing stable-kf5-qt5 SUSEQt5.9	1 mo 12 days - #2	N/A	29 min
Applications kdenlive kf5-qt5 FreeBSDQt5.9	2 days 2 hr - #2	N/A	9 min 57 sec
Applications kdenlive kf5-qt5 SUSEQt5.9	2 days 2 hr - #7	1 mo 15 days - #1	15 min
Applications kdenlive kf5-qt5 WindowsMSVCQt5.9	N/A	2 mo 22 days - #8	59 min
Applications kdenlive stable-kf5-qt5 FreeBSDQt5.9	2 days 1 hr - #3	N/A	18 min
Applications kdenlive stable-kf5-qt5 SUSEQt5.9	2 days 1 hr - #14	N/A	3 min 51 sec
Applications kdenlive stable-kf5-qt5 WindowsMSVCQt5.9	N/A	2 days 1 hr - #22	21 min
Applications kdepim-addons kf5-qt5 SUSEQt5.9	9 hr 25 min - #108	N/A	37 min
Applications kdepim-addons stable-kf5-qt5 SUSEQt5.9	11 days - #31	N/A	42 min

Starting of Jobs

- SCM, Software configuration management:
  (`* /5 * * * *` – checking new revision(s))
- Timer, cron-like (`00 23 * * *` – nightly build)
- Executed manually
- After another job's finish – pipeline

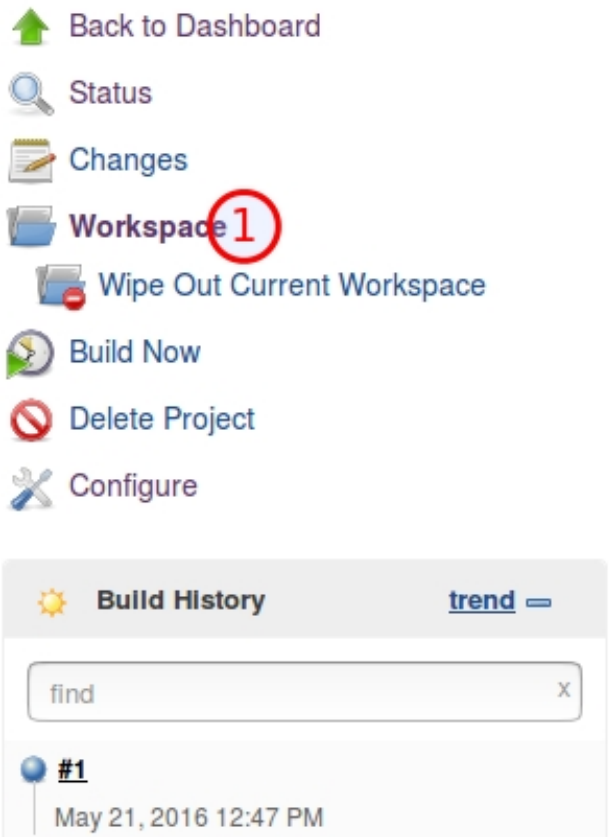
Build Info


SUSEQt5.9 #5
Changes
1. GIT_SILENT: Time to increase version (detail / cgit)
Started by an SCM change
Revision: 41bda0c3efc2315c981a260a40c471761a210313
origin/master
Revision: 139f68b528771a15a9e1394abd89da09969dda97
origin/master
Revision: 9413af76df3d6bb10b1da0565ec19f6d996903a4
origin/master
Revision: 191224048912b0d5dcb1f9750a01ae6176502137
origin/master
Revision: a5a402aac02b5dc8846c6ee3d82e592548af3001
origin/master
Revision: 4e8c366aa5157182e797446d503afe268d27ae74
origin/master
Test Result (1 failure / ±0)
GNU Make + GNU C Compiler Warnings: 5 warnings.

Console log

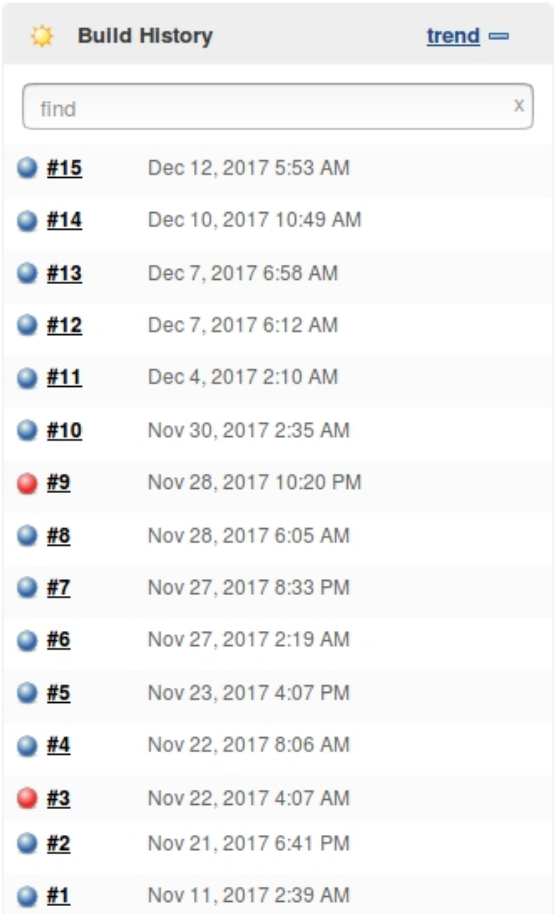
```
03:25:43 -- Build files have been written to: /home/jenkins/workspace/Applications bovo stable-kf5-qt5 SUSEQt5.9/build
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Compiling)
[Pipeline] sh
03:25:43 [Applications bovo stable-kf5-qt5 SUSEQt5.9] Running shell script
03:25:43 + python3 -u ci-tooling/helpers/compile-build.py --product Applications --project bovo --branchGroup stable-
kf5-qt5 --platform SUSEQt5.9 --usingInstall /home/jenkins//install-prefix/
03:25:43 Scanning dependencies of target bovo_autogen
03:25:43 [  3%] Generating settings.h, settings.cpp
03:25:43 [  6%] Automatic MOC for target bovo
03:25:44 Generating MOC predefs moc_predefs.h
03:25:44 Generating MOC source AQUW3KFNJO/moc_ai.cpp
03:25:44 Generating MOC source AQUW3KFNJO/moc_aifactory.cpp
03:25:44 Generating MOC source TTHINB22PQ/moc_aiaron.cpp
03:25:44 Generating MOC source OJNGYDLLRA/moc_aigabor.cpp
03:25:44 Generating MOC source ZL6H5ELRUJ/moc_game.cpp
03:25:44 Generating MOC source DMHXEJ42XS/moc_hintitem.cpp
03:25:44 Generating MOC source DMHXEJ42XS/moc_mainwindow.cpp
03:25:44 Generating MOC source DMHXEJ42XS/moc_mark.cpp
03:25:44 Generating MOC source DMHXEJ42XS/moc_scene.cpp
03:25:44 Generating MOC compilation mocs_compilation.cpp
03:25:44 [  6%] Built target bovo_autogen
03:25:44 Scanning dependencies of target doc-index-cache-bz2
03:25:44 [ 10%] Generating index.cache.bz2
03:25:44 Scanning dependencies of target bovo
03:25:44 [ 13%] Building CXX object CMakeFiles/bovo.dir/gui/hintitem.cc.o
03:25:44 [ 16%] Building CXX object CMakeFiles/bovo.dir/gui/mark.cc.o
03:25:44 [ 20%] Building CXX object CMakeFiles/bovo.dir/gui/theme.cc.o
03:25:45 Note: namesp. add : added namespace before processing              The Bovo Handbook
03:25:45 [ 23%] Building CXX object CMakeFiles/bovo.dir/gui/view.cc.o
03:25:45 [ 23%] Built target doc-index-cache-bz2
03:25:45 [ 26%] Building CXX object CMakeFiles/bovo.dir/gui/mainwindow.cc.o
```

Workspace


Back to Dashboard
Status
Changes
Workspace
1
Wipe Out Current Workspace
Build Now
Delete Project
Configure
Build History
trend
find
#1
May 21, 2016 12:47 PM


Workspace of FirstJob-CreateFile on master
script1.txt 13 B view
2
script2.txt 19 B view
(all files in zip)

Build History



Changes

Changes

Summary

1. GIT_SILENT Upgrade KDE Applications version to 17.11.80. (details)
2. ArkViewer: close on Escape (details)
3. ExtractJob: set destination URL in the job description (details)

Commit 510d93dcda38195857a6e945079a20b825ce81ae **by** Albert Astals Cid
```
GIT_SILENT Upgrade KDE Applications version to 17.11.80.
```

CMakeLists.txt (diff)

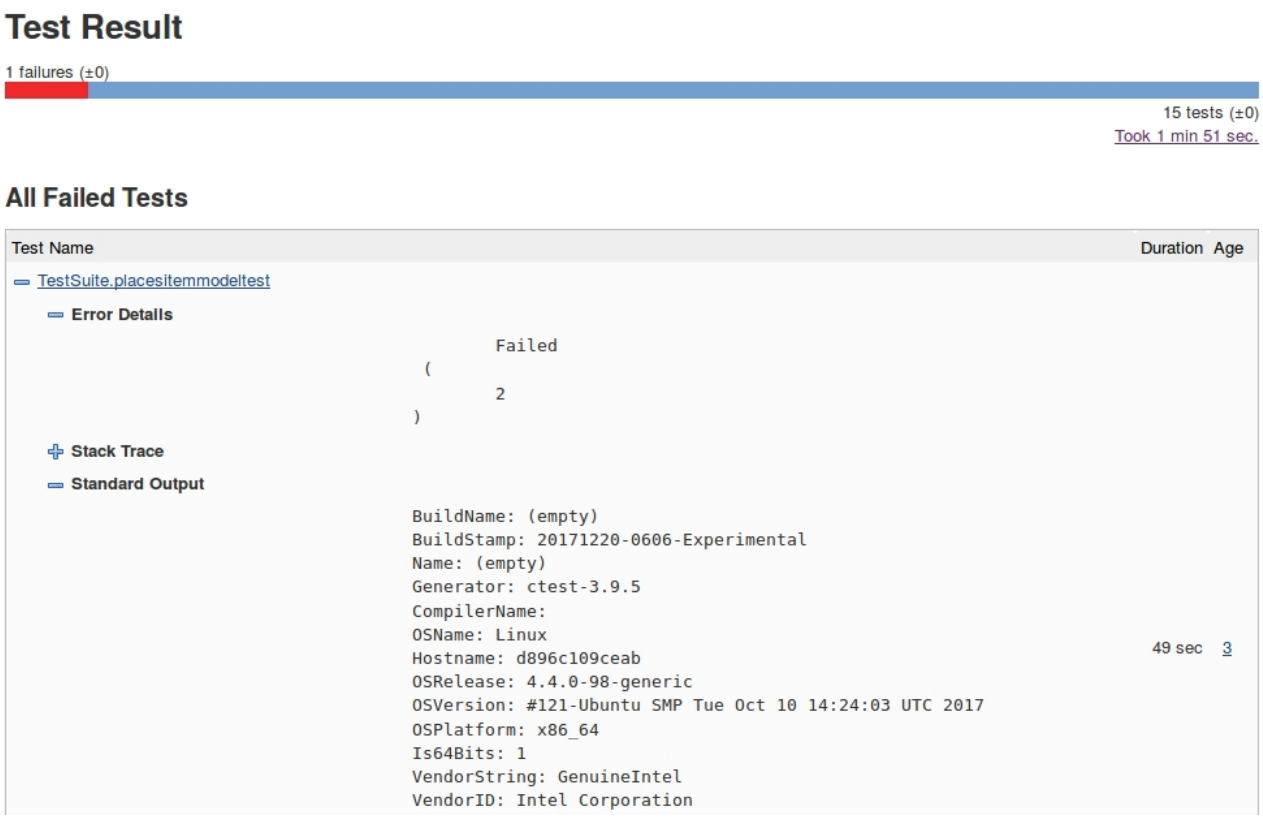
Commit ae758226b8b630572de715054b4b965443a4ef11 **by** Elvis Angelaccio
```
ArkViewer: close on Escape
BUG: 386729 FIXED-IN: 17.11.80
Test Plan: Open preview for a text file, then check whether Esc closes
the window. If some operation is happening in the kate part (e.g. text
selected, or search bar open), make sure Esc aborts the operation rather
than closing the window.
Reviewed By: broulik
Differential Revision: D8761
```

part/arkviewer.cpp (diff)

Commit 52cbbd803a2f57a971372fde1b5ebcc3808bfb7a **by** Elvis Angelaccio
```
ExtractJob: set destination URL in the job description
The plasma notification applet looks for it for the "Open" button within
the notification. If it doesn't find it, it falls back to the archive
URL which is not what one would expect.
BUG: 385043 FIXED-IN: 17.11.90
Differential Revision: D8861
```

kerfuffle/jobs.cpp (diff)

Test Result



Test Result

1 failures (±0)

15 tests (±0)

Took 1 min 51 sec.

All Failed Tests

Test Name	Duration	Age
TestSuite.placesitemmodeltest		
Error Details		

```
        Failed
 (
        2
)
```

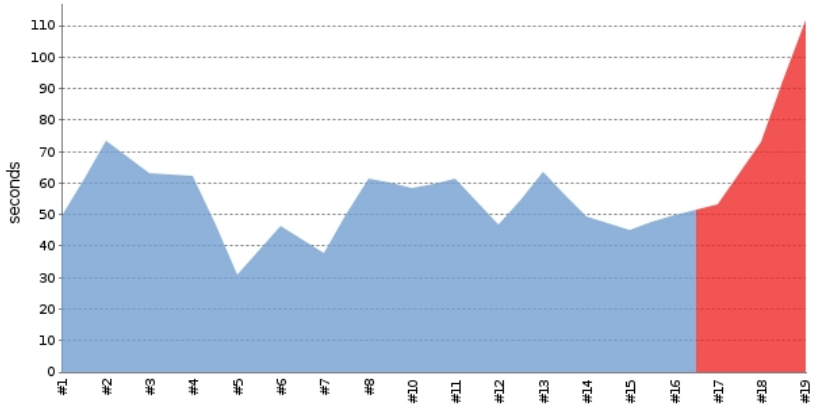
Stack Trace		
Standard Output		

```
BuildName: (empty)
BuildStamp: 20171220-0606-Experimental
Name: (empty)
Generator: ctest-3.9.5
CompilerName:
OSName: Linux
Hostname: d896c109ceab
OSRelease: 4.4.0-98-generic
OSVersion: #121-Ubuntu SMP Tue Oct 10 14:24:03 UTC 2017
OSPlatform: x86_64
Is64Bits: 1
VendorString: GenuineIntel
VendorID: Intel Corporation
```

49 sec 3

Test History

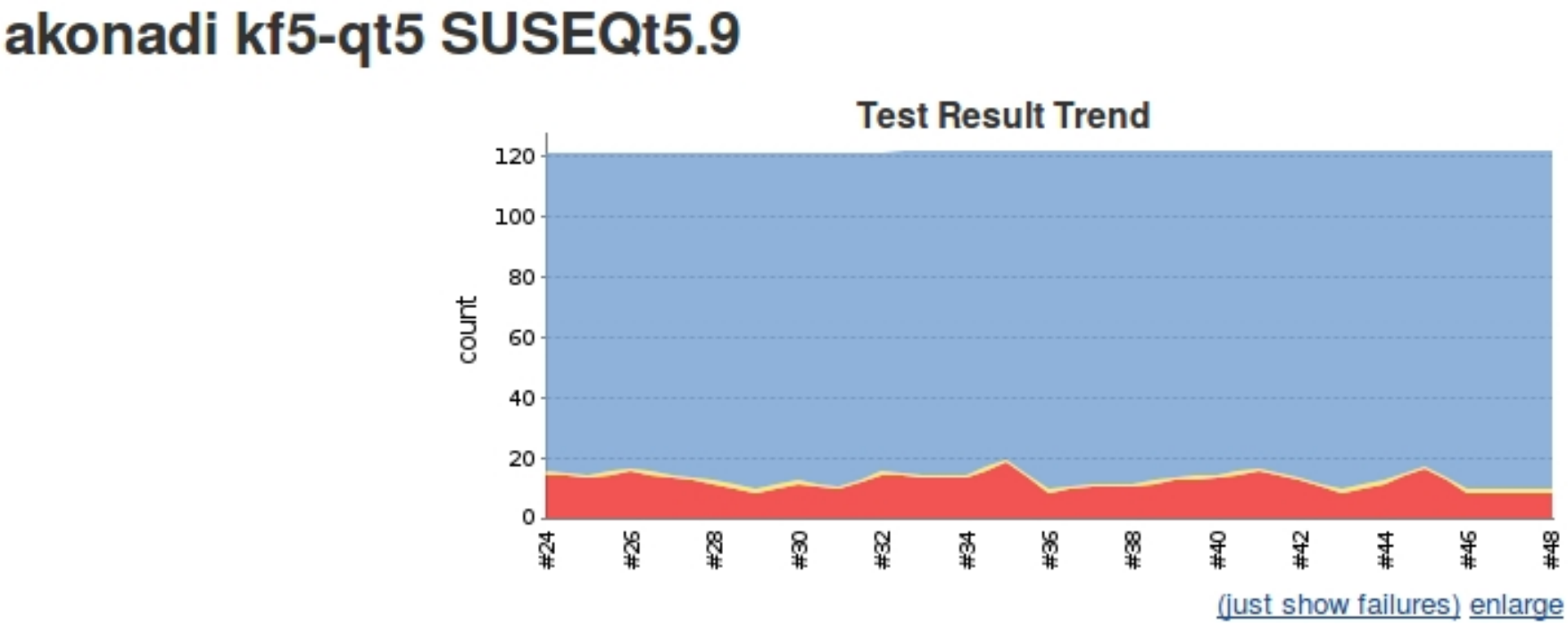
History for Test Results



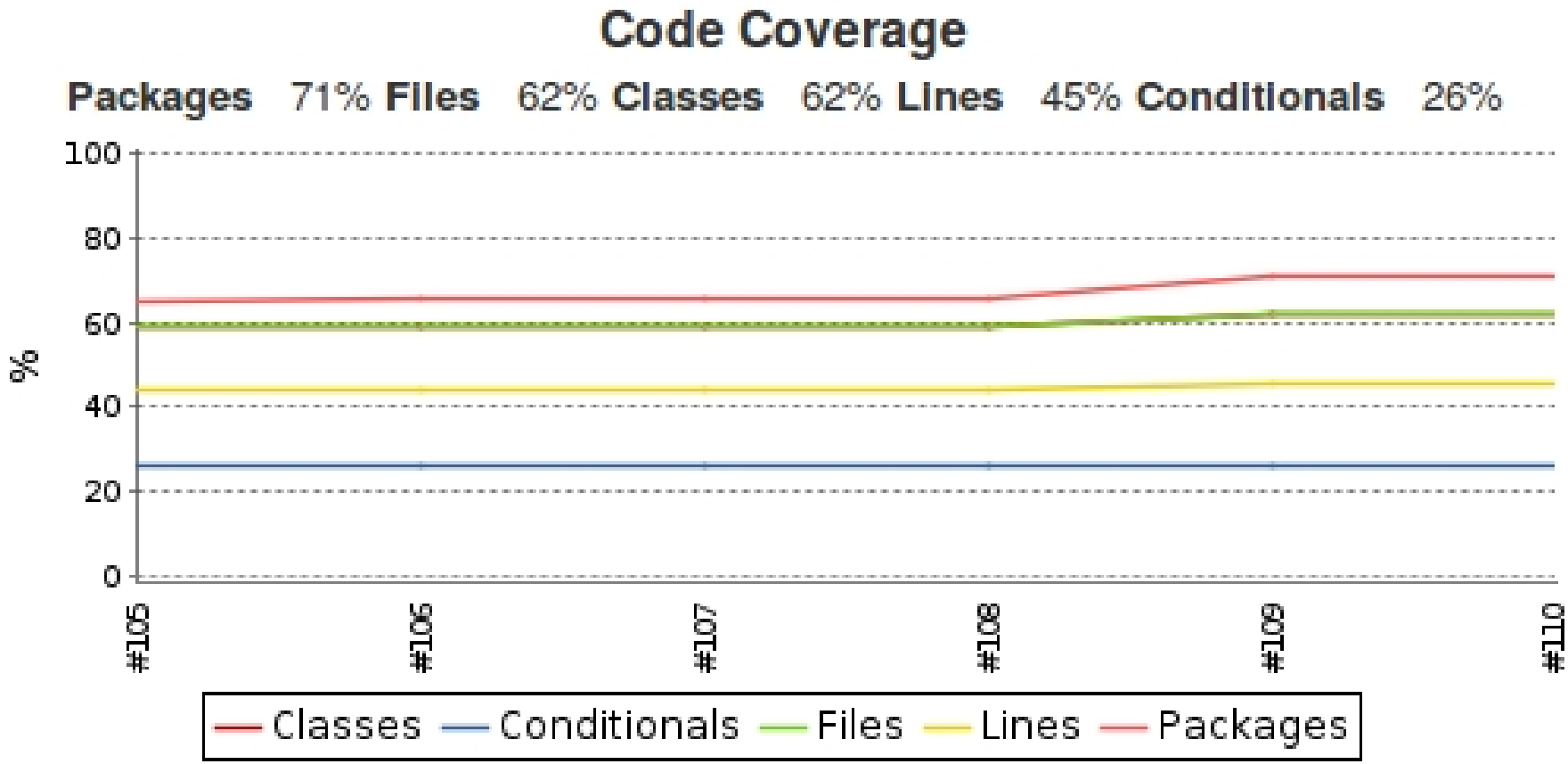
show count

Build	Description	Duration	Fail	Skip	Total
Applications dolphin kf5-qt5 SUSEQt5.9 #19		1 min 51 sec	1	0	15
Applications dolphin kf5-qt5 SUSEQt5.9 #18		1 min 12 sec	1	0	15
Applications dolphin kf5-qt5 SUSEQt5.9 #17		53 sec	1	0	15
Applications dolphin kf5-qt5 SUSEQt5.9 #16		49 sec	0	0	15
Applications dolphin kf5-qt5 SUSEQt5.9 #15		44 sec	0	0	15
Applications dolphin kf5-qt5 SUSEQt5.9 #14		49 sec	0	0	15
Applications dolphin kf5-qt5 SUSEQt5.9 #13		1 min 3 sec	0	0	15
Applications dolphin kf5-qt5 SUSEQt5.9 #12		46 sec	0	0	15
Applications dolphin kf5-qt5 SUSEQt5.9 #11		1 min 1 sec	0	0	14
Applications dolphin kf5-qt5 SUSEQt5.9 #10		58 sec	0	0	14

Test Result Trend



Code Coverage



Pipeline – Stage View

Jenkins 2: Declarative/Scripted (Groovy) pipeline jobs

Applications kcalutils stable-kf5-qt5 SUSEQt5.9 - Stage View

	Checkout Sources	Setup Dependencies	Configuring Build	Compiling	Installing	Capturing Installation	Running Tests	Checking Code Quality
Average stage times:	35s	36s	9s	35s	1s	10s	20s	3s
#10 Dec 15 07:58 — 1 commits	18s	42s	12s	53s	1s	7s	17s	1s
#9 Dec 04 11:30 — No Changes	41s	14s	13s	55s	2s	9s	20s	1s
#8 Dec 04 11:20 — 1 commits	22s	14s	15s	43s	1s	6s	18s	1s
#7 Dec 02 21:02 — 1 commits	2min 10s	41s	7s	42s	2s	35s	30s	13s
#6 Nov 23 15:30 — 1 commits	17s	41s	2s	19s	874ms	5s	14s	1s
#5 Nov 21 20:17 — No Changes	24s	22s	13s	46s	872ms	9s	15s	1s

Reports

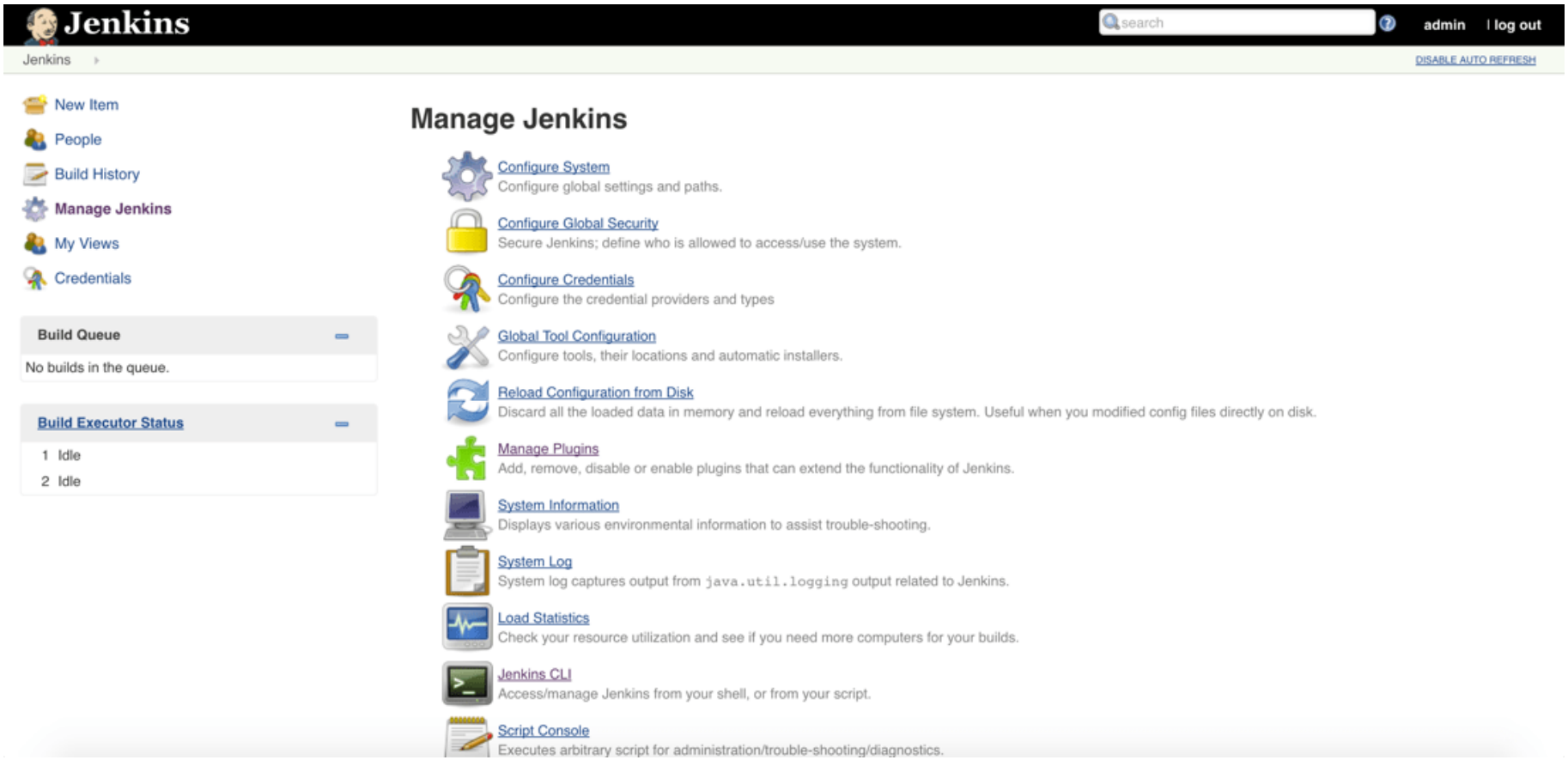
Many reporting options can be used:

- Jenkins UI
- E-mail
- Slack notification
- etc.

Creating New Jobs

- UI support
- Jobs are stored in xml files (e.g. `/var/lib/jenkins/jobs`)
- The jobs can be stored in version control
- Jenkins reload can be triggered by updating the version controlled job(s)

The Configuration Dashboard



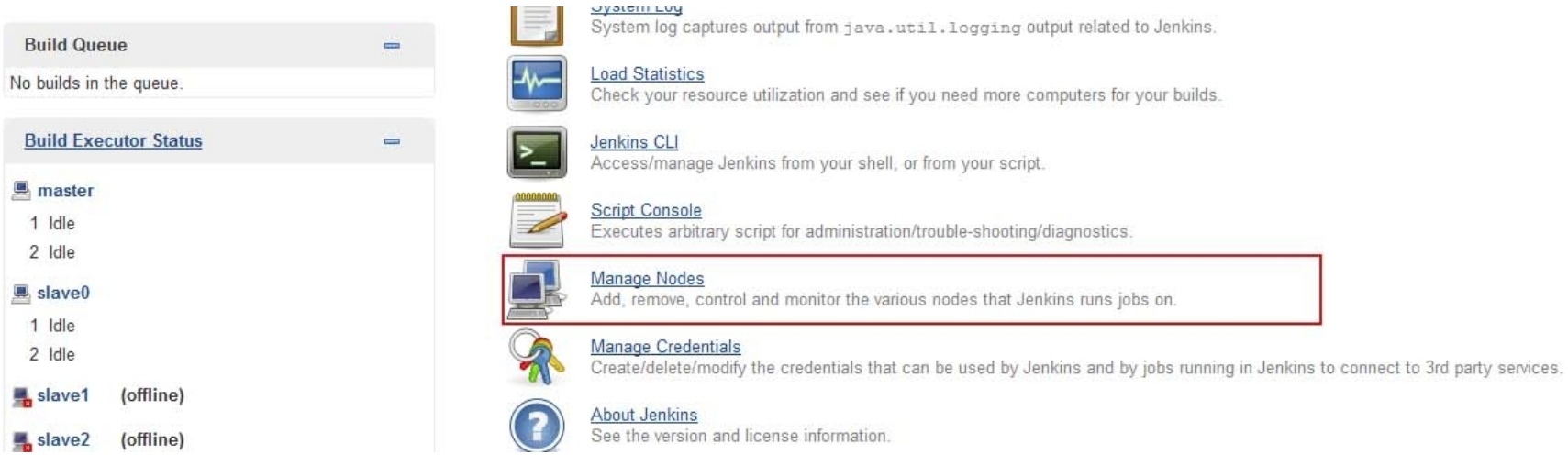
Configuration

- Number of executors
- Number of jobs, tests split into jobs
- Length of build history per jobs

Configuration

Master - Slave nodes:

- Reduced CPU, memory and I/O load
- Parallel execution
- High Availability
- May using different Operating Systems



Plugins

More hundreds plugins are available:

- SCM: svn, perforce, git, mercurial, clear case, etc.
- shell, batch execution
- Building: MSBuild, CMake, ant, maven, etc.
- Testing: CppTest, JUnit, Selenium, etc.
- Static analyzers: Cppcheck, CheckStyle, etc.
- Code coverage: gcov, Clover, etc.
- Others: scp, radiator, lava lamp, etc.

Conclusion

- CI – Supporting the software development in teamwork
- Avoid Integration hell
- CI Principles and tools
- Jenkins: widely-used CI server with many useful plugins
- Jenkins Web UI:
  - Jobs
  - Console log
  - Previous builds
  - Test running
  - Trends
  - etc.

Useful Links

- Martin Fowler: Continuous Integration: `https://martinfowler.com/articles/continuousIntegration.html`
- Darko Fabijan: Why We Need Continuous Integration: `https://semaphoreci.com/community/tutorials/continuous-integration`
- Jenkins: `https://jenkins-ci.org/`
- Jenkins pipelines: `https://jenkins.io/2.0/#pipelines`

Thank You

Thank you for your attention!