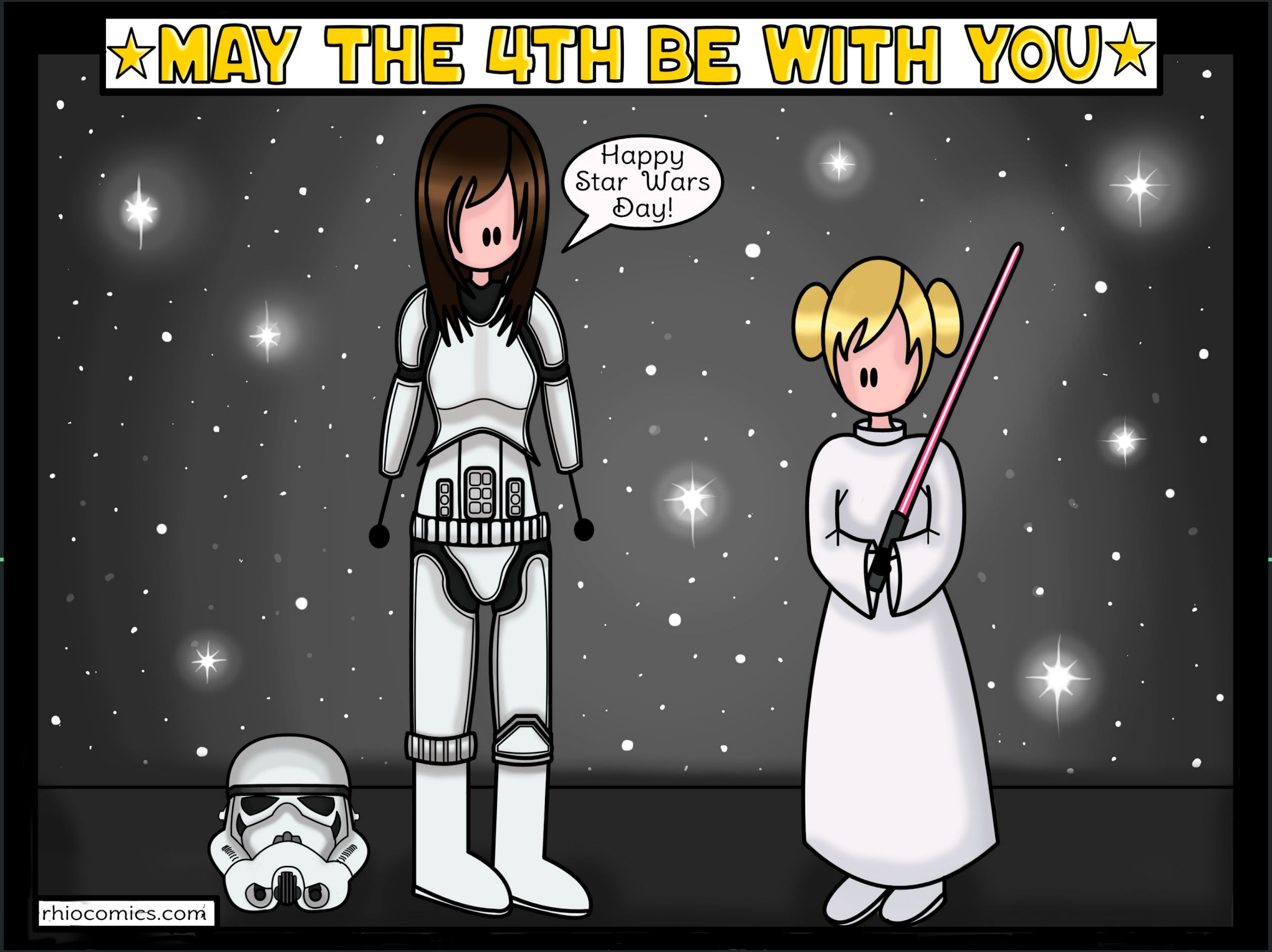
Continuous Integration

Pete Bartram & Sam Diserens


★MAY THE 4TH BE WITH YOU★
Happy Star Wars Day!
rhiocomics.com

Outline

- Introduction
- Overview of Jenkins
- Setting up Jenkins
- Jenkins Workshop

Introduction

- What is Continuous Integration (CI)
- Why do we use CI
- CI in Commercial Software Development
- Options for Continuous Integration

What is Continuous Integration

- *"Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early."*

  - ThoughtWorks

- Continually introduce small incremental changes to the code base
- Changes are tested as they are applied
- Can be combined with version control software, e.g. GIT, SVN

Why Continuous Integration?

- Identify conflicts early
  - Multiple developers contributing to a shared codebase
- Reduce time taken to find and fix bugs
  - Bug in 10 vs 1,000 lines of code?
- Automation of testing
  - Scheduled daily testing
  - Release testing
- Pin-point which change (who) broke the code

Why Continuous Integration?

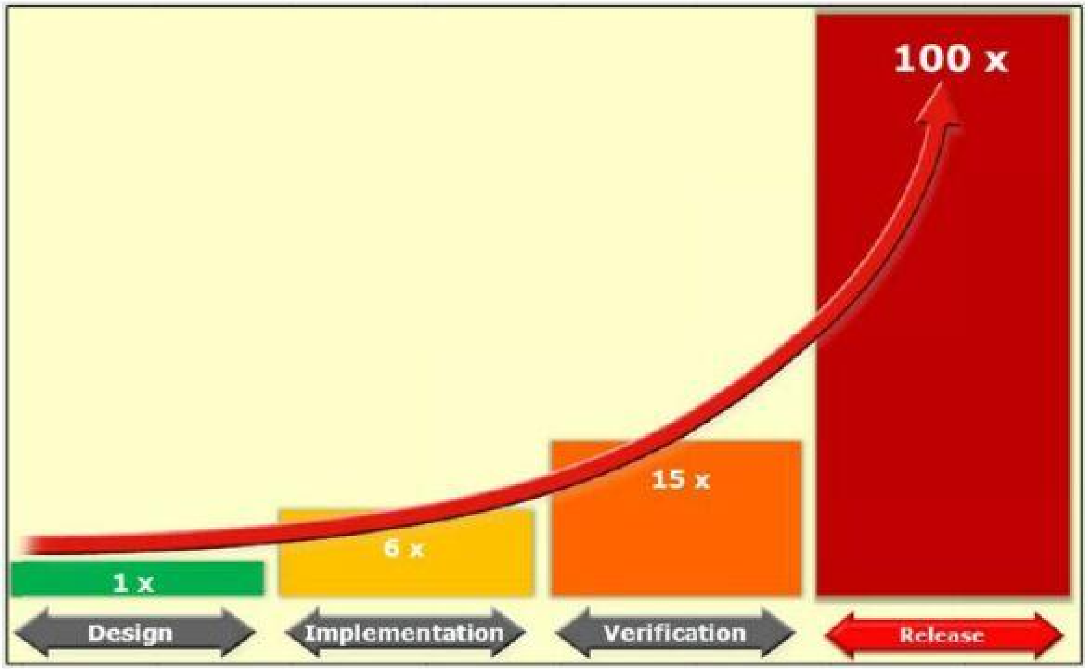


Figure 1: Cost of Bug Elimination in the Software Development Lifecycle [NIST 2002]

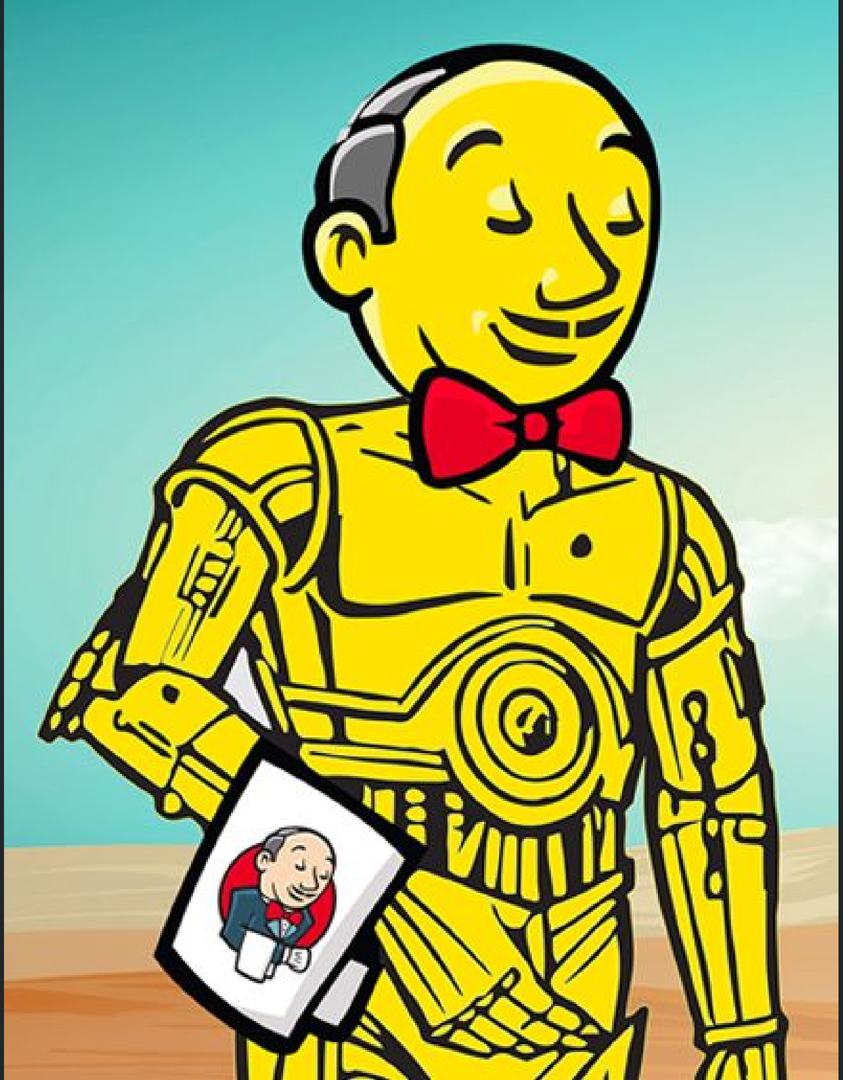
Continuous Integration in the wild

- 20+ Developers and QAs
- Version Control System, e.g. GiT
- Build tests triggered on commit
  - Email notifications on failed build
- Test suites run overnight
  - Displays showing passing/failing tests

Options for Continuous Integration

- Jenkins/Hudson
  - Most popular free CI tool
- (Hudson)
  - Original name for Jenkins
  - Continues to be run by Oracle
- Build Bot
  - Written in python
  - Flexible and lightweight
- Cruise Control
  - One of the the first available CI solutions (2001 release)
- Team Foundation Server (TFS)
  - Microsoft’s offering
  - Combines project/requirements management as well as CI tools

Jenkins

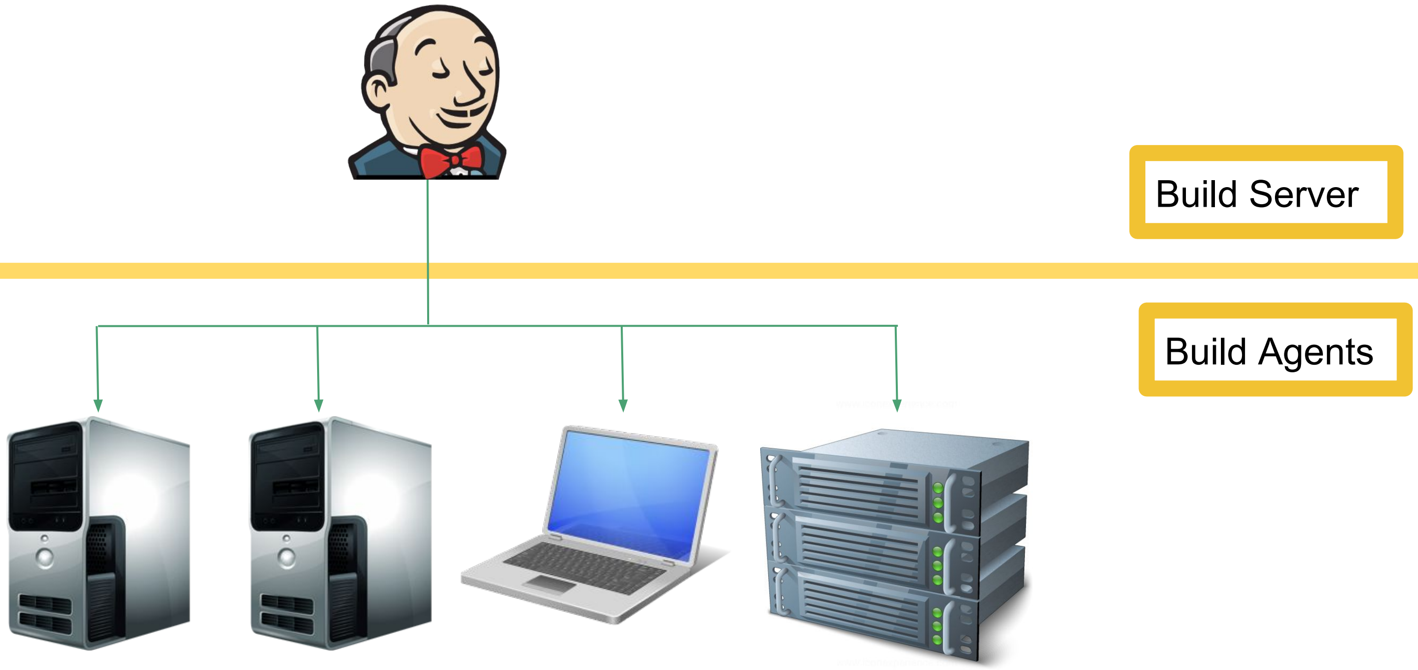


What is Jenkins?

Web server based tool

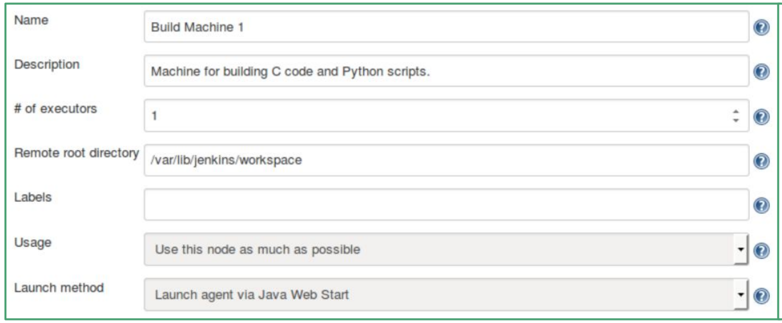
- Builds code
- Parses for warnings / errors
- Runs unit tests
- Displays results
- Generates documentation
- Handles code releases

Jenkins Server / Client Model

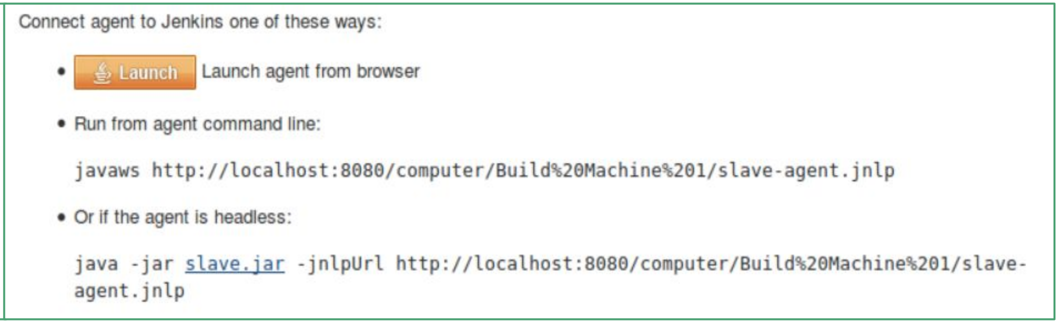


Jenkins Server and Client - Initial Setup

- Server setup is straightforward: https://wiki.jenkins-ci.org/display/JENKINS/Installing+Jenkins+on+Ubuntu
- Client Configuration less so:
  - Create a new node: Dashboard -> Manage Jenkins -> Manage Nodes -> New Node
  - Configure Jenkins agent.
  - sudo apt-get install icedtea-netx
  - Start the agent: Dashboard -> Manage Jenkins -> Manage Nodes -> Node Name



Name	Build Machine 1
Description	Machine for building C code and Python scripts.
# of executors	1
Remote root directory	/var/lib/jenkins/workspace
Labels	
Usage	Use this node as much as possible
Launch method	Launch agent via Java Web Start



Connect agent to Jenkins one of these ways:

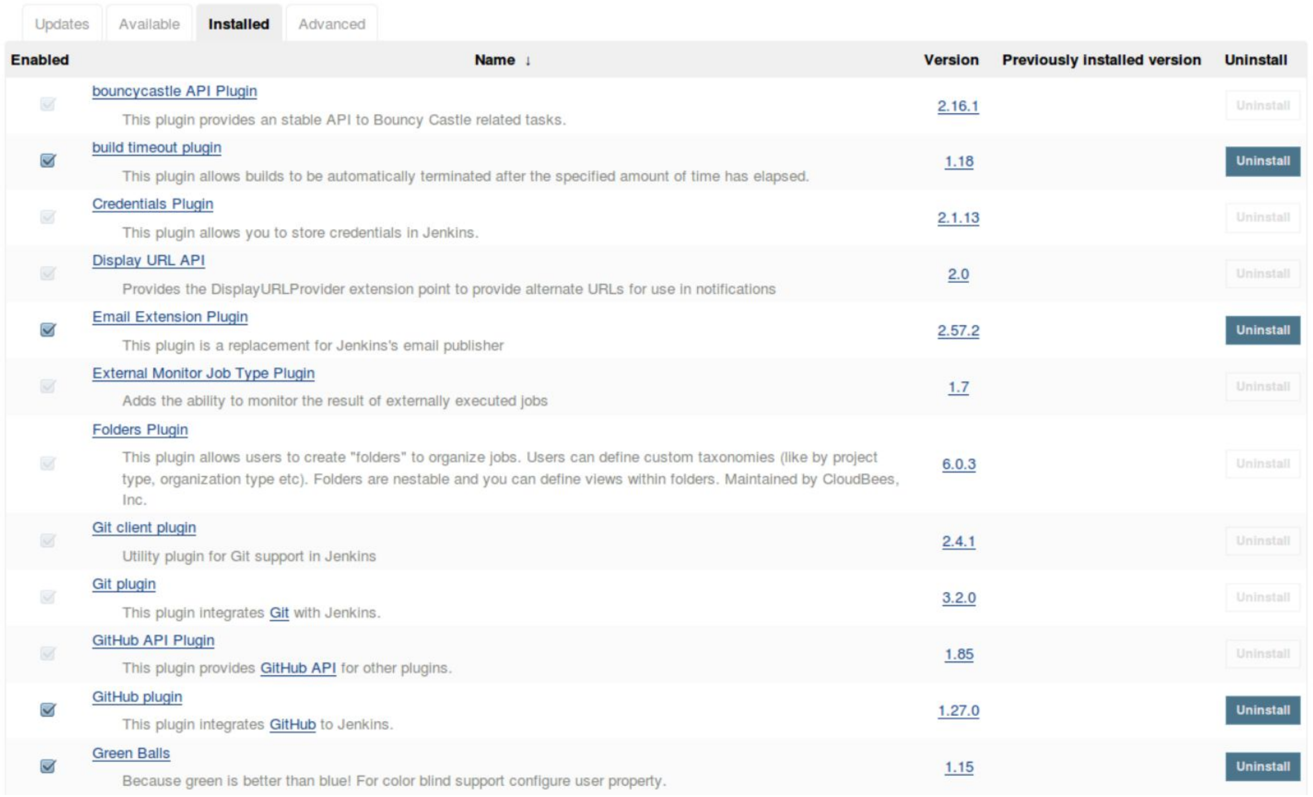
- Launch — Launch agent from browser
- Run from agent command line:

```
javaws http://localhost:8080/computer/Build%20Machine%201/slave-agent.jnlp
```

- Or if the agent is headless:

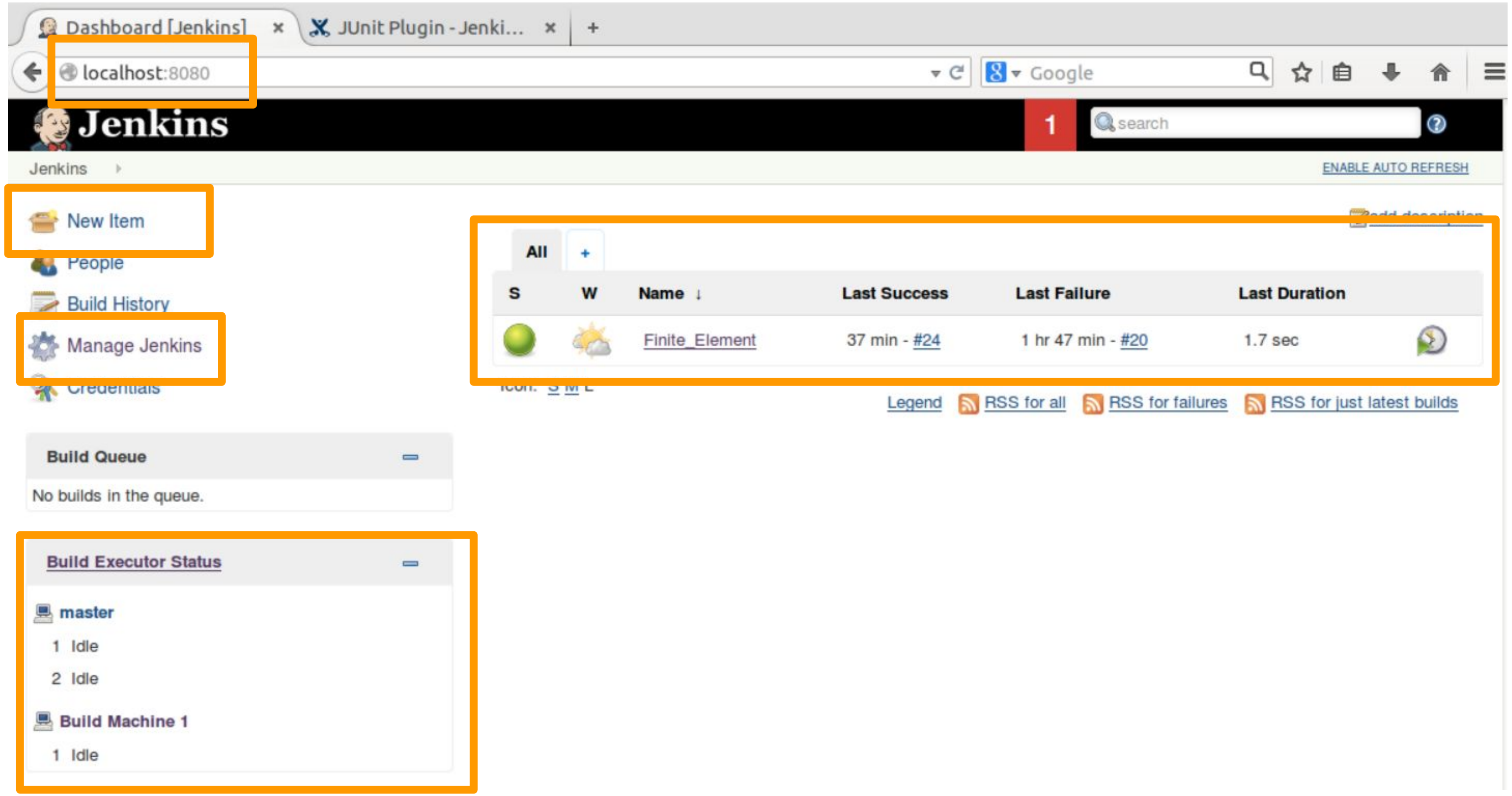
```
java -jar slave.jar -jnlpUrl http://localhost:8080/computer/Build%20Machine%201/slave-agent.jnlp
```

Plugin Management


Updates | Available | **Installed** | Advanced

Enabled	Name ↓	Version	Previously installed version	Uninstall
✓	bouncycastle API Plugin This plugin provides an stable API to Bouncy Castle related tasks.	2.16.1		Uninstall
✓	build timeout plugin This plugin allows builds to be automatically terminated after the specified amount of time has elapsed.	1.18		Uninstall
✓	Credentials Plugin This plugin allows you to store credentials in Jenkins.	2.1.13		Uninstall
✓	Display URL API Provides the DisplayURLProvider extension point to provide alternate URLs for use in notifications	2.0		Uninstall
✓	Email Extension Plugin This plugin is a replacement for Jenkins's email publisher	2.57.2		Uninstall
✓	External Monitor Job Type Plugin Adds the ability to monitor the result of externally executed jobs	1.7		Uninstall
✓	Folders Plugin This plugin allows users to create "folders" to organize jobs. Users can define custom taxonomies (like by project type, organization type etc). Folders are nestable and you can define views within folders. Maintained by CloudBees, Inc.	6.0.3		Uninstall
✓	Git client plugin Utility plugin for Git support in Jenkins	2.4.1		Uninstall
✓	Git plugin This plugin integrates Git with Jenkins.	3.2.0		Uninstall
✓	GitHub API Plugin This plugin provides GitHub API for other plugins.	1.85		Uninstall
✓	GitHub plugin This plugin integrates GitHub to Jenkins.	1.27.0		Uninstall
✓	Green Balls Because green is better than blue! For color blind support configure user property.	1.15		Uninstall

Jenkins Dashboard Tour



Dashboard [Jenkins] | JUnit Plugin - Jenki...

localhost:8080

Jenkins

Jenkins › ENABLE AUTO REFRESH

- New Item
- People
- Build History
- Manage Jenkins
- Credentials

add description

All | +

S	W	Name ↓	Last Success	Last Failure	Last Duration
●	⛅	Finite_Element	37 min - #24	1 hr 47 min - #20	1.7 sec

Icon: S M L

Legend | RSS for all | RSS for failures | RSS for just latest builds

Build Queue

No builds in the queue.

Build Executor Status

master

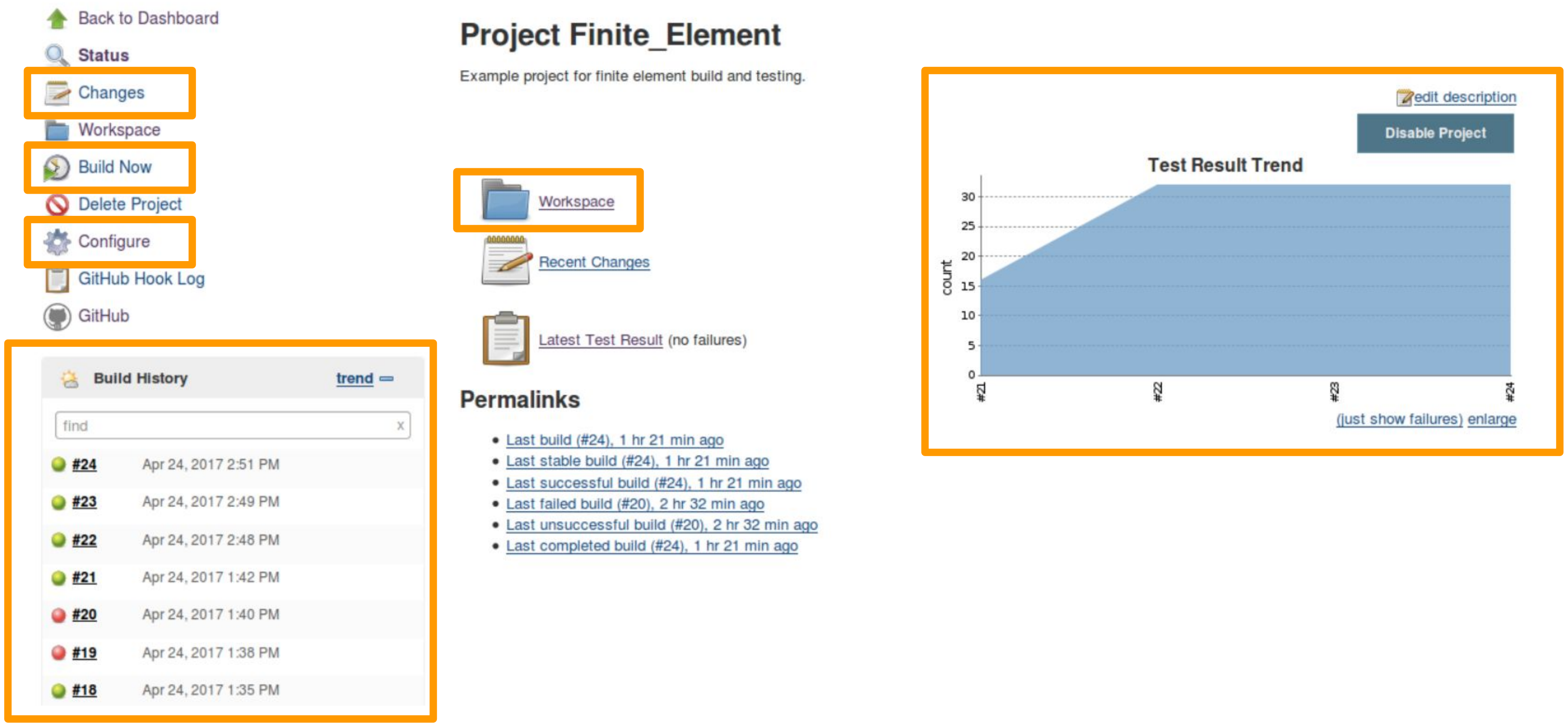
1 Idle

2 Idle

Build Machine 1

1 Idle

Project Home Page



Back to Dashboard
Status
Changes
Workspace
Build Now
Delete Project
Configure
GitHub Hook Log
GitHub

Build History trend

find

#24	Apr 24, 2017 2:51 PM
#23	Apr 24, 2017 2:49 PM
#22	Apr 24, 2017 2:48 PM
#21	Apr 24, 2017 1:42 PM
#20	Apr 24, 2017 1:40 PM
#19	Apr 24, 2017 1:38 PM
#18	Apr 24, 2017 1:35 PM

Project Finite_Element

Example project for finite element build and testing.

Workspace

Recent Changes

Latest Test Result (no failures)

Permalinks

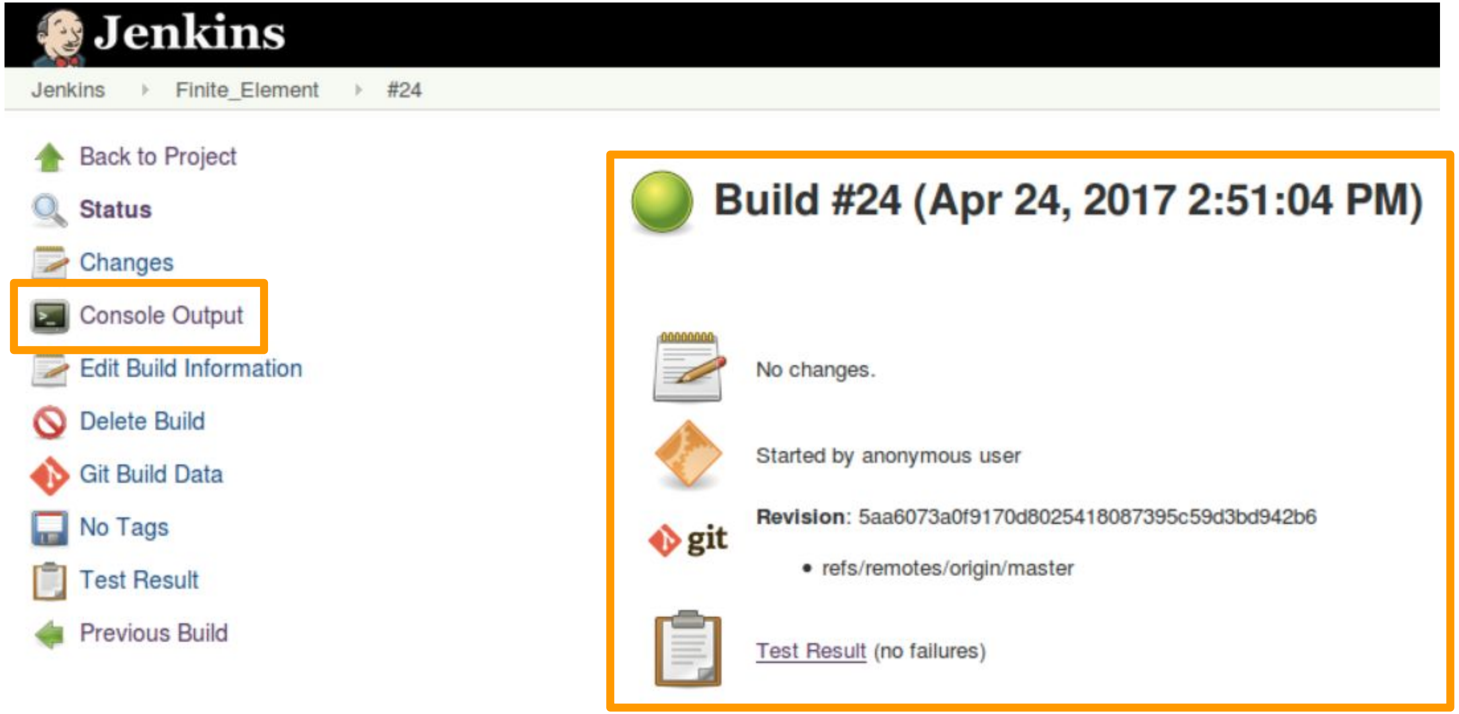
- Last build (#24), 1 hr 21 min ago
- Last stable build (#24), 1 hr 21 min ago
- Last successful build (#24), 1 hr 21 min ago
- Last failed build (#20), 2 hr 32 min ago
- Last unsuccessful build (#20), 2 hr 32 min ago
- Last completed build (#24), 1 hr 21 min ago

edit description

Disable Project

(just show failures) enlarge

Build Page



Jenkins

Jenkins ‣ Finite_Element ‣ #24

- Back to Project
- **Status**
- Changes
- Console Output
- Edit Build Information
- Delete Build
- Git Build Data
- No Tags
- Test Result
- Previous Build

Build #24 (Apr 24, 2017 2:51:04 PM)

No changes.

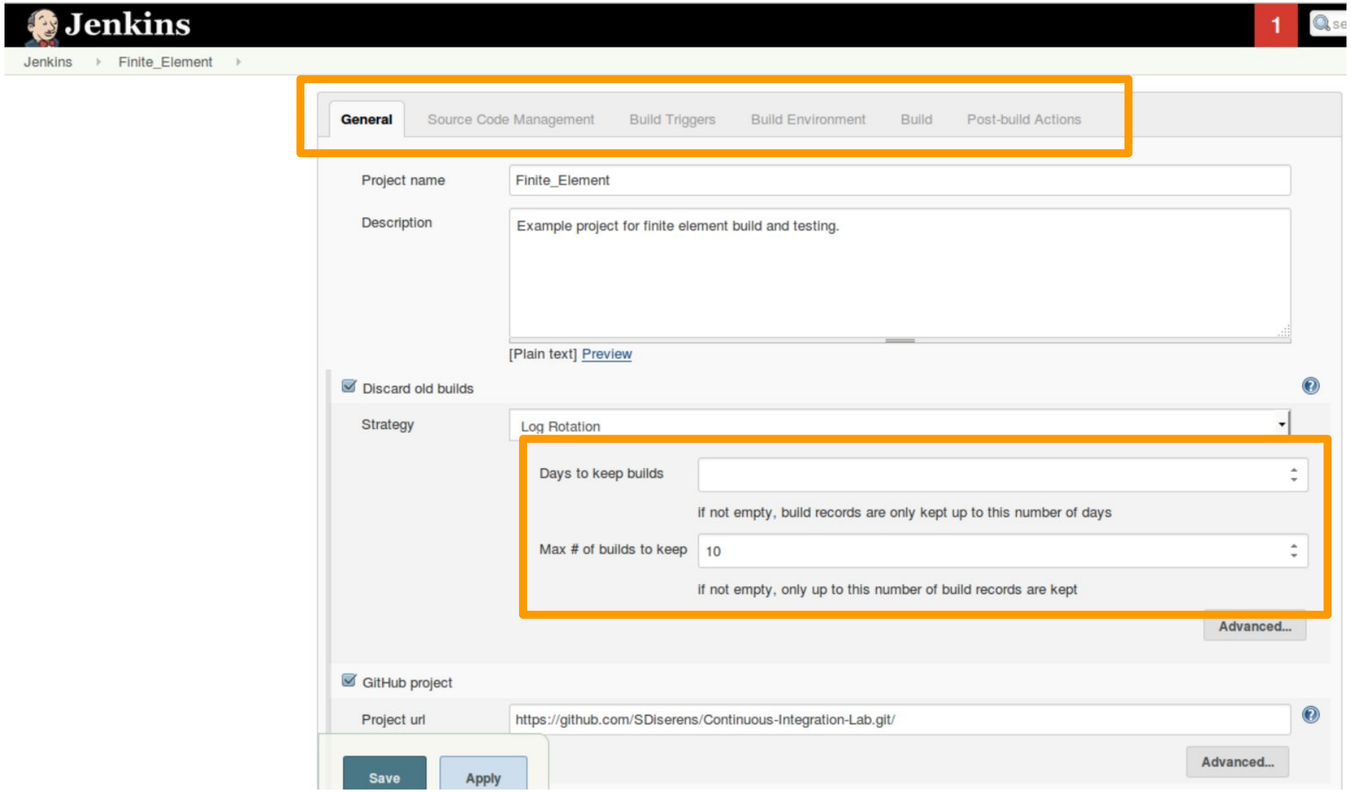
Started by anonymous user

Revision: 5aa6073a0f9170d8025418087395c59d3bd942b6

- refs/remotes/origin/master

Test Result (no failures)

Project Configuration Page



Project name: Finite_Element

Description: Example project for finite element build and testing.

[Plain text] Preview

✓ Discard old builds

Strategy: Log Rotation

Days to keep builds:

if not empty, build records are only kept up to this number of days

Max # of builds to keep: 10

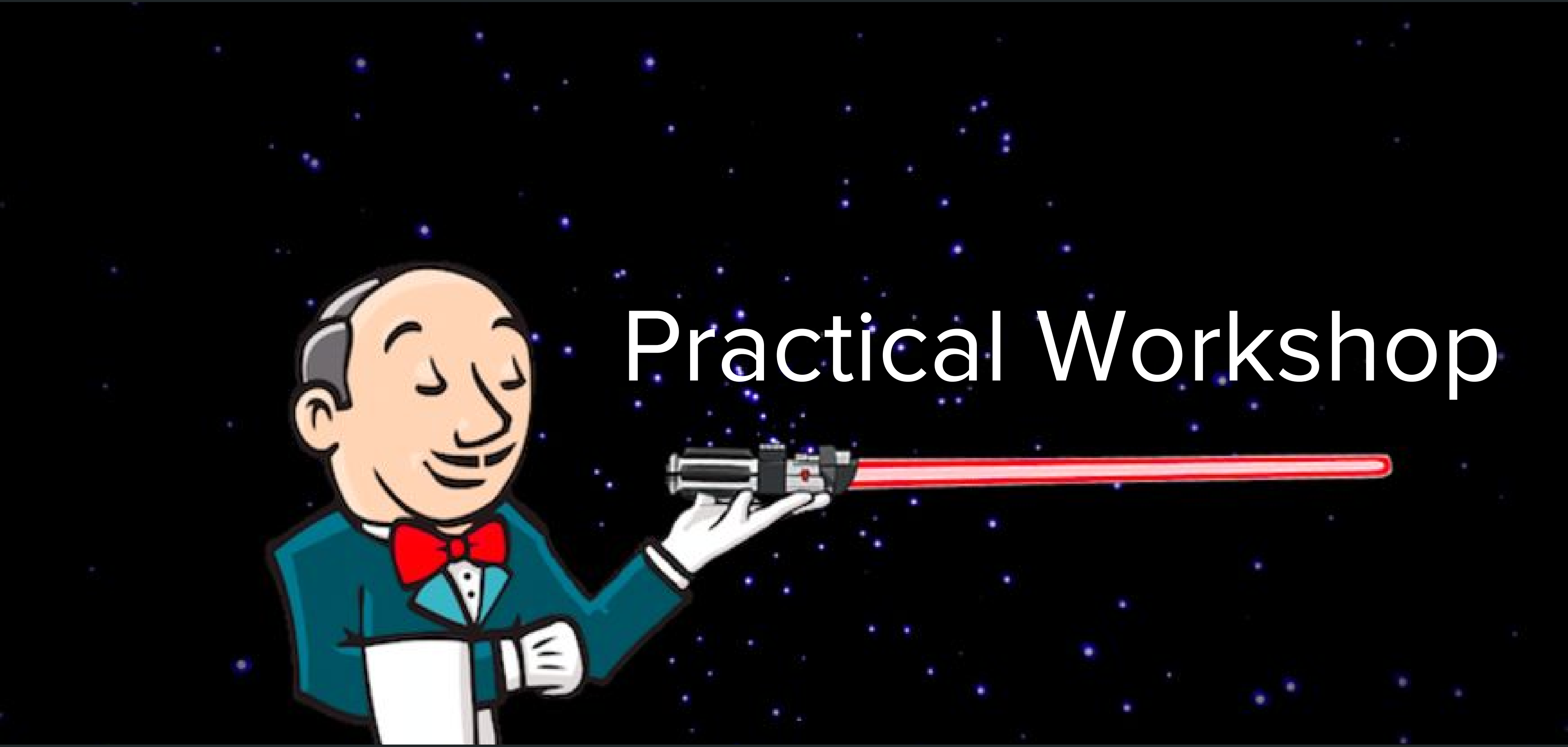
if not empty, only up to this number of build records are kept

✓ GitHub project

Project url: https://github.com/SDiserens/Continuous-Integration-Lab.git/

Breaking The Build

"We have a full-size Justin Bieber cutout that gets placed facing the team member who broke the build. We found that 100% of software engineers don't like Justin Bieber, and will work quickly to fix the build problem." - Robert Rose, SpaceX Lead Avionic Software Engineer.



Practical Workshop

Summary

- Bad practices
  - Polling of GitHub every minute.
  - Not splitting jobs across multiple builds.
- Managing more complex jobs with a build script
- In reality we might implement:
  - Code run on commit hook
    - Tests the integration of the change
    - Know immediately if something is broken
  - Run tests at specific times
    - Tests the functionality of the software
    - Can be scheduled to run overnight when user load is less

Thank You!

FOR NO APPARENT REASON: DARTHFIELD
DARTHFIELD, WE'RE OUT OF LASAGNA.
URK
ooK
I HATE MONDAYS!
LUCAS