Build Pipeline Plugin

Summary

This plugin provides a *Build Pipeline View* of upstream and downstream connected jobs that typically form a build pipeline. In addition, it offers the ability to define manual triggers for jobs that require intervention prior to execution, e.g. an approval process outside of Jenkins.

Plugin Information
View Build Pipeline on the plugin site for more information.

Release Notes

1.5.8

- Address boundary case of JENKINS-23532 where relative paths would not retrigger

1.5.7.1

- (-) JENKINS-45137 Fix stack trace for pipelines created / edited in v 1.5.6

1.5.7

- (-) JENKINS-23532 Manual trigger execution causes TriggerException
- (-) JENKINS-44324 NullPointerException upgrading from very old versions of plugin
- (-) Fix issue with dashboard view not loading
- (+)Added support for extensible BuildCard

1.5.6

- Issues #34722 (Pull #88) Performance fix when determining view permissions
- Issues #14591 (Pull #96 and Pull #106) Add the ability to choose which parameters build will be displayed

1.5.5

- Eaten by INFRA-588

1.5.4

- Pull #105 Keep git-plugin RevisionParameterAction on rerun of a build
- Issues #31088 show displayName by default

1.5.3.1

- Issue #33935 Fix re-running a build - broken in Jenkins > 1.653, LTS > 1.651.1
- Issue #34722 Performance fix - limit the number of downstream projects iterated over

1.5.2

- Issue #33591 Fix manual trigger breaking in Jenkins >1.653

1.5.1

- Issue #31746 Fix issues in layout, UI updates

1.4.9

- (issue #30801) Re-triggering a failed build copies the Actions from previous builds
- (issue #28068) Build Pipeline Dashboard View destroys layout of jenkins
- (issue #29477) View bad for Build Pipeline Plugin

1.4.8

- (issue #28180) Build Pipeline background layout does not extend full width of pipeline

1.4.7

- (JENKINS-25666) Fixed left-side indentation for new jenkins versions

1.4.6

- (JENKINS-20499) Null cause brokes Cause entry and job
- (JENKINS-22665) (JENKINS-19755) Don't store whole user object for cause
- Fixed UI tests

- Removed test libs from final package

1.4.5

- Support cloudbees-folder-plugin (JENKINS-14565) (JENKINS-20841)
- Start build with parameters for parametrized builds (JENKINS-25427) (JENKINS-19121)
- Clicking on "console" icon doesn't work (JENKINS-25430)

1.4.4

1.3.3
1.3.1
1.3.0 - Also see the roadmap for details.
1.2.4 - Also see the roadmap for details.
1.2.2
1.2.1
1.2
1.1.2
1.1.1
1.0.0

Overview

Continuous Integration has become a widely adopted practice in modern software development. Jenkins & Hudson are great tools for supporting Continuous Integration.

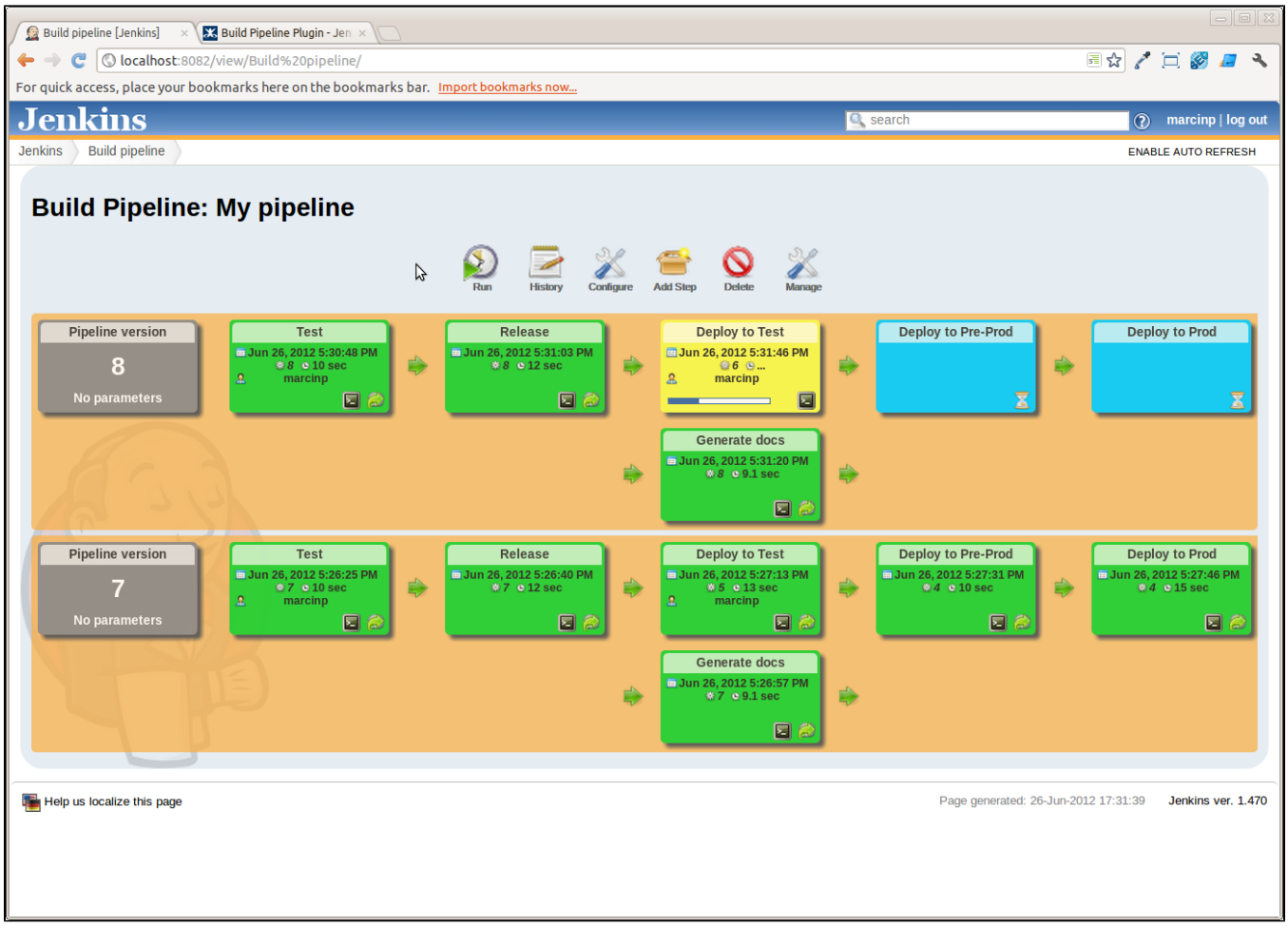
Taking it to the next level: Continuous integration can become the centerpiece of your deployment pipeline, orchestrating the promotion of a version of software through quality gates and into production. By extending the concepts of CI you can create a chain of jobs each one subjecting your build to quality assurance steps. These QA steps may be a combination of manual and automated steps. Once a build has passed all these, it can be automatically deployed into production.

In order to better support this process, we have developed the Build Pipeline Plugin. This gives the ability to form a chain of jobs based on their upstream\downstream dependencies. Downstream jobs may, as per the default behaviours, be triggered automatically ,or by a suitable authorised user manually triggering it.

You can also see a history of pipelines in a view, the current status and where each version got to in the chain based on its revision number in VCS.

Screenshots

The Pipeline View



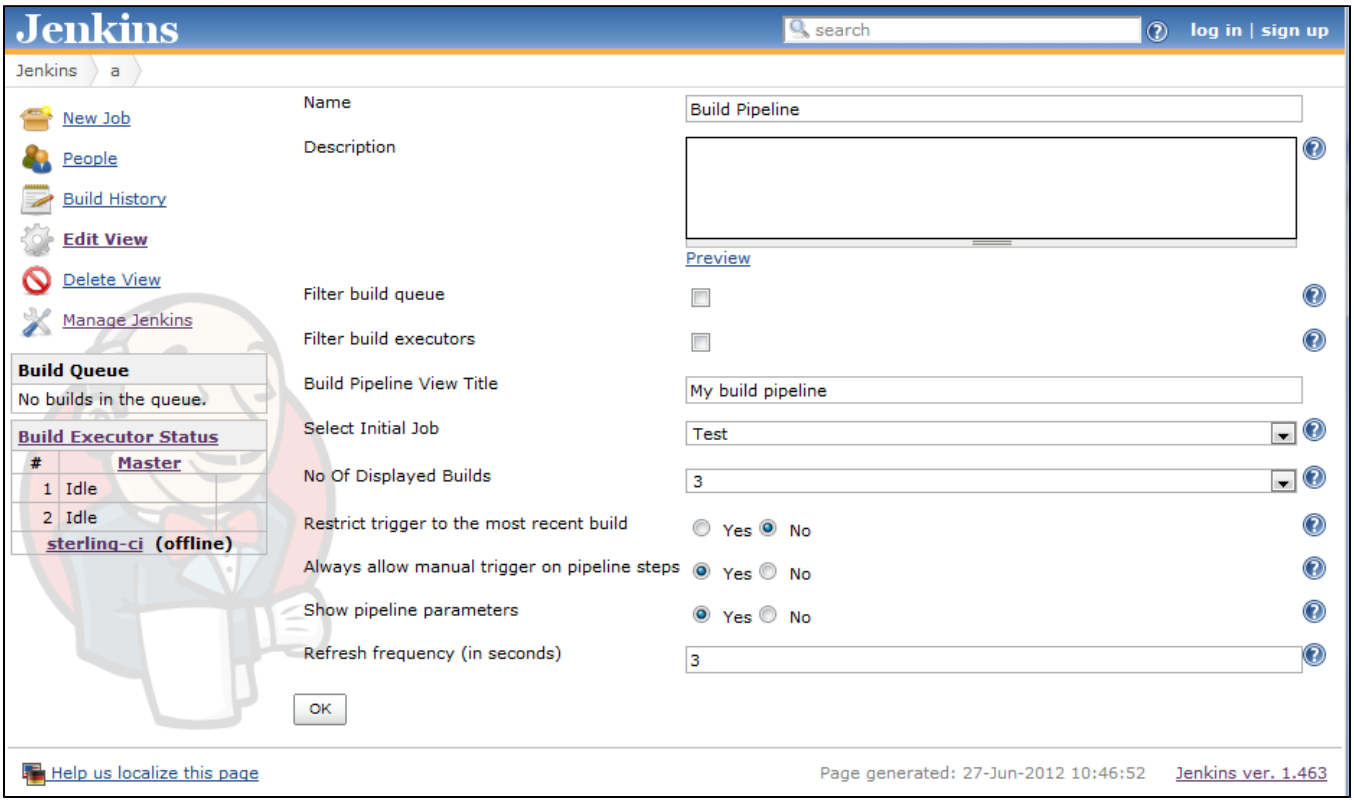
Configuration

View Configuration

1. Install the plugin using the Hudson\Jenkins Plugin Manager and restart.
2. Create a view of the new type *Build Pipeline View*.
   You will then be redirected directly to the configuration page.
3. The table below outlines what each interesting parameter controls:

Name	The name of the Build Pipeline View
Description	This message will be displayed on the view page. Useful for describing what this view is about, or linking to relevant resources. Can contain HTML tags.
Build Pipeline View Title	Gives a title to the page that displays the view
Select Initial Job	This is the first job in the build pipeline. It will traverse through the downstream jobs to build up the entire build pipeline. Select from a drop-down list of jobs.
No of Displayed Builds	The number of historical builds to be displayed on a page.
Restrict triggers to most recent successful builds	Select this option to restrict the display of a Trigger button to only the most recent successful build pipelines. *Yes:* Only the most recent successful builds displayed on the view will have a manual trigger button for the next build in the pipeline. *No:* All successful builds displayed on the view will have a manual trigger button for the next build in the pipeline.
Always allow manual trigger on pipeline steps	Select this option if you want to manually execute or re-execute any step of the pipeline at any time.

Show pipeline parameters	Select this option if you want to display the parameters used to run the first job in the pipeline.

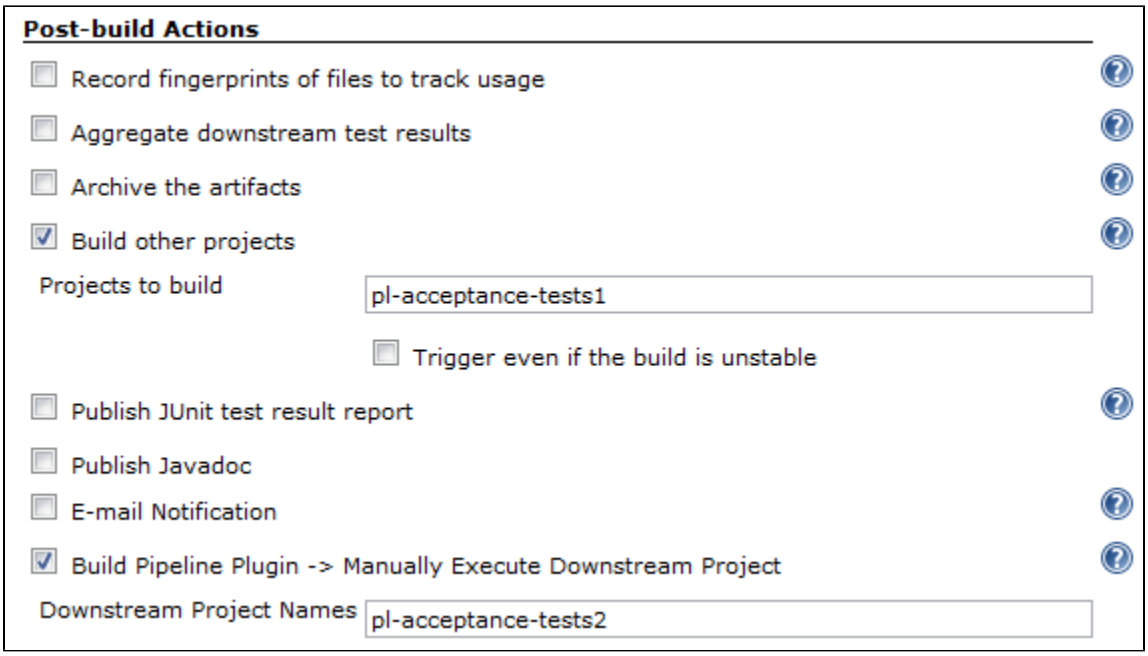


Job Configuration

1. Navigate to the Job configuration page.
2. Scroll down to the *Post-build Actions* section.
    a. For an **Automated** downstream build step;
    To add a build step that will trigger automatically upon the successful completion of the previous one:
        i. Select the *Build other projects* check-box
        ii. Enter the name(s) of the downstream projects in the *Projects to build* field. (n.b. Multiple projects can be specified by using comma, like "abc, def".)
    b. For a **Manually Triggered** downstream build step:
    To add a build step that will wait for a manual trigger:
        i. Select the *Build Pipeline Plugin -> Manually Execute Downstream Project* check-box
        ii. Enter the name(s) of the downstream projects in the *Downstream Project Names* field. (n.b. Multiple projects can be specified by using comma, like "abc, def".)
3. Click *Save*

Automatic & Manual downstream build steps

The Build Pipeline Plugin handles the creation of multiple automatic and/or manually triggered downstream build steps on the same project.



Upgrading from Release 1.0.0

When upgrading from 1.0.0 to 1.1.x some of the previous view and job configuration fields have been removed. You may notice some errors of the following errors appearing in the Hudson/Jenkins log.

```
WARNING: Skipping a non-existent field downstreamProjectName
com.thoughtworks.xstream.converters.reflection.NonExistentFieldException: No such field
au.com.centrumsystems.hudson.plugin.buildpipeline.trigger.BuildPipelineTrigger.downstreamProjectName
```

This is because the configuration files refer to old fields that may no longer exist.
In order to correct these issues go to the Job configuration page, confirm that all of the details are correct and click on the *Save* button.

More on Pipelines

The canonical reference for pipelines is the book Continuous Delivery.

Chapter 5 of the book, which describes how deployment pipelines work, is available for free here.