

# Pipeline Stage View Plugin

## Plugin Information

View Pipeline: Stage View [on the plugin site](#) for more information.



This plugin is up for adoption. Want to help improve this plugin? [Click here to learn more!](#)

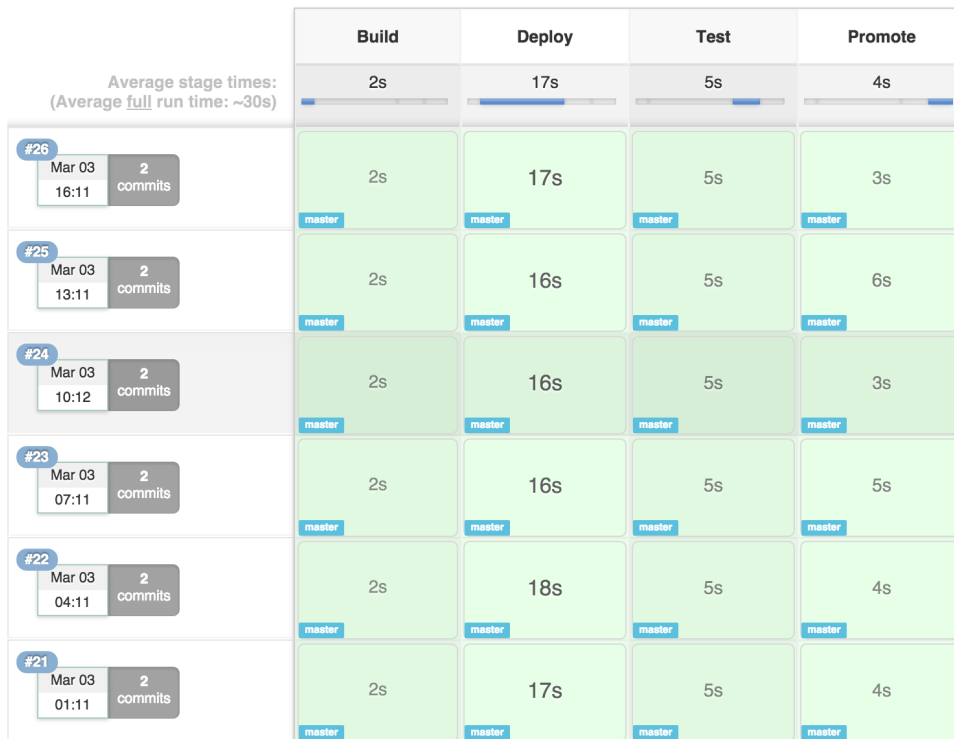
## Stage View:

### Stage View

Average stage times:  
(Average full run time: ~1min 19s)

	Build docker images	Unit Test ubuntu-python27	Unit Test centos6-python26	Unit Test debian-wheezy using python-3.4.3
#140 Mar 02 08:51 No Changes	5s	32s failed		
#139 Mar 02 08:17 No Changes	4s	30s	12s	31s
#138 Mar 02 00:22 No Changes	4s	30s	12s	31s
#137 Mar 01 21:42 No Changes				

## Stage View



When you have complex builds Pipelines, it is useful to be able to see the progress of each stage. The Pipeline Stage View plugin includes an extended visualization of Pipeline build history on the index page of a flow project, under *Stage View*. (You can also click on *Full Stage View* to get a full-screen view.)

To take advantage of this view, you need to define stages in your flow. You can have as many stages as you like, in a linear sequence. (They may be inside other steps such as 'node' if that is convenient.) Give each a short name that will appear in the GUI.

```
node {
  stage ('Checkout') {
    svn 'https://svn.mycorp/trunk/'
    stage 'Build'
    sh 'make all'
    stage 'Test'
    sh 'make test'
  }
}
```

When you run some builds, the stage view will appear with *Checkout*, *Build*, and *Test* columns, and one row per build. When hovering over a stage cell you can click the *Logs* button to see log messages printed in that stage: after the preceding stage step and before the next one.

Other important events are also indicated with special buttons, either in a stage cell or for the build as a whole. If a build is waiting in an input step, you will be able to tell it to proceed (or not). *Download* can be used to obtain any artifacts archived by a build.

Each row also records the build number, date it was started, and any changelog entries from a version control system. Progress bars also indicate how long each stage is taking and how long it might still be expected to take, based on historical averages.

## Usage Notes/Limits:

- Block-scoped stages (stages that take a closure, see below) may occur in parallels but are not displayed in parallel.

```
stage('name'){ echo 'steps here' }
```

- Dynamic stages:** in general, if you want to visualize dynamically changing stages, make it conditional to execute the stage contents, not conditional to **include** the stage\*\* Stage view can handle a limited subset of cases where you're appending additional stages, but the cell-based view is not matched to changing stage structures
- Configurable limits are provided for the REST API that drives stage view -- these may be overridden though (see below).
- The pipeline stage view uses caching internally for performance: even if the first viewing is slow, following ones should be quite fast

## Configurable Properties:

- At runtime or startup, you may turn off Jenkins user lookup for changesets with property `com.cloudbees.workflow.rest.external.ChangeSetExt.resolveCommitAuthors`
  - This is a workaround for rare cases where you are relying on an external security realm for user information and have performance issues with many calls:
  - In the script console this setting may be changed at runtime (with immediate impact):

```
System.setProperty("com.cloudbees.workflow.rest.external.ChangeSetExt.resolveCommitAuthors", "false");
```

- To turn user lookup back on:

```
System.setProperty("com.cloudbees.workflow.rest.external.ChangeSetExt.resolveCommitAuthors", "true");
```

- Hardcoded API limits that may be overridden by setting the properties at startup (requires restarting Jenkins to see the change):
  - Characters in each step's log entry (default: 10240 or 10kB) - `com.cloudbees.workflow.rest.external.FlowNodeLogExt.maxReturnChars`
  - Runs displayed per job (default: 10) - `com.cloudbees.workflow.rest.external.JobExt.maxRunsPerJob`
  - Artifacts shown per run (default: 100) - `com.cloudbees.workflow.rest.external.RunExt.maxArtifactsCount`
  - Steps displayed per stage (default: 100) - `com.cloudbees.workflow.rest.external.StageNodeExt.maxChildNodes` - note: this may have a very large performance impact with complex builds.

## Changelog:

### 2.12 (August 6, 2019)

- Fix Internet Explorer 11 compatibly [JENKINS-58808](#)

### 2.11 (April 19, 2019)

- Avoid generating NaNy NaNd (in case of Nan)
- Add chinese translation
- Fix overlap when used with Test Result Trend
- Link to full node log from the window
- Improve usability of the download artefacts popup

### 2.10 (Mar 30, 2018)

- Fix a small bug with URL generation that could result in invalid URL generation in rare cases with customized Jenkins hostnames

### 2.9 (Aug 29, 2017)

- Fix a timing bug for run durations - [JENKINS-40162](#)
- Fix Jackson 2 dependencies by switching to use the Jenkins plugin rather than the library [JENKINS-46526](#)
- Fix an issue with Stage View caching where job renames generated bad links - [JENKINS-38159](#)

### 2.8 (May 25, 2017)

- **Major feature:** Display arguments used in running steps in the logs view - [JENKINS-37324](#)
- Fix handling of BooleanParameter default to false - [JENKINS-36543](#)

### 2.6 (Mar 6, 2017)

- [JENKINS-38536](#) Fix timing of incomplete parallels

### 2.5 (Feb 13, 2017)

- Fix a minor CSS layout issue with stage view interfering with test results trends graphs

### 2.4 (Nov 29, 2016)

- Fix: Regression introduced in 2.3 which resulting in requesting all runs for a job, even if most were cached [JENKINS-40087](#)

### 2.3 (Nov 22, 2016)

- Fix: Stage View showed multiple rows if displayName changed for a run (community PR, thank you!) - [JENKINS-34538](#)

## 2.2 (Nov 2, 2016)

- CSS Fix: excess whitespace on sides of stages table with very high resolution monitors

## 2.1 (Sept 30, 2016)

- Fix: Prevent errors in deeply nested blocks from overflowing the stage status window ([JENKINS-37945](#))
  - Also filters out block start/end steps from that and log display (reverts to 1.7 behavior)
- Fix: hide NotExecuted steps from log display (user is resuming from checkpoint, these steps never ran on this run)
- CSS Fix: prevent excessively long error messages from overflowing stage status popup
- Feature: display timing information for steps in log view
- Don't show log-expand icons if there is no log message for a step ([JENKINS-31833](#))

## 2.0 (Aug 30, 2016)

- Feature: support visualization of block-scoped stages (labelled blocks) ([JENKINS-26107](#))
- **Major feature:** complete rewrite of the flow analysis algorithms ([JENKINS-34038](#))
  - Needed to support block-scoped stage, and originally intended to allow for visualizing parallels
  - Scalability: simpler and more efficient APIs allow for analyzing flows in a logical one-pass fashion.
  - Performance: allows for far greater performance. Currently roughly at parity with 1.7, but once [JENKINS-37569](#) is done, should be \*much\* faster.
  - Optimization: includes one final round of caching improvements. We can now serve requests about individual stages/nodes from the cache, and caches use less memory.

## 1.7 (Aug 6, 2016)

- Performance/bugfix: massively improve serverside run/stage caching, by greatly limiting excess cache invalidation
  - Cache no longer fully invalidates on renaming a pipeline job: instead the old cache entries are removed and new ones created with the new job name
  - Bugfix: Need to invalidate run cache results when their job is deleted ([JENKINS-36865](#))
  - Known issue: run cache can no longer serve requests for data about individual stages/nodes.
    - Mitigation: UI switches to use API below, so it just gets this data with the run request
- Performance: Fetch full stage data when requesting runs, and doesn't issue separate requests for individual stages ([JENKINS-34036](#))
  - API extension to allow getting full stage data for requested runs (or just the stage minimal data without child nodes)
  - Front-end (browser) uses this API now
  - It will still issue requests for in-progress builds though (room for some additional work)
  - Clientside stage cache is redone to cache all completed stages when request issued.
- Minor fix to CSS for unstable build ([JENKINS-37001](#))

## 1.6 (July 18, 2016)

Theme: Status handling and UX

- Stage view correctly handles try/catch blocks ([JENKINS-34212](#))
  - Workaround: stages with errors are marked successful if overall build is.
- Better support use of currentBuild.result to modify run status ([JENKINS-36523](#) offers one example)
- Support use of the UNSTABLE status for runs ([JENKINS-33700](#))
  - Catch: you can only set this status for the whole build (result is that successful stages will instead show as unstable).
  - Needs [JENKINS-26522](#) to provide custom status tracking for a stage.
- Fix bizarre timing numbers from stage view ([JENKINS-34644](#))
- Fix Stage View jumping to the left when displaying many stages and scrolled to the right ([JENKINS-36384](#))
- Allow multi-line input prompts ([JENKINS-36661](#))

## 1.5 (June 21, 2016)

Theme: configurability of hardcoded values

- REMOVE node labels in stage view (per [JENKINS-34038](#)) until we can correctly handle more of the cases with a block-based pipeline graph analysis
- Add properties to override hardcoded limits in the REST API (i.e. to show more or less builds, artifacts, etc) ([JENKINS-34791](#))
- Add a property (modifiable at runtime) to disable lookup of Jenkins users for commit changelogs ([JENKINS-35484](#))
  - Useful for cases where user lookup ties to a security realm and caching problems or network calls render it expensive.

## 1.4 (May 12, 2016)

- Ensure the log view popup is always placed in the visible area of the page ([JENKINS-33109](#))
- Make full screen mode with use the full width by removing offsets for sidebar ([JENKINS-33498](#))
- Fix display of artifact names when there are large numbers ([JENKINS-33351](#))
- Fix an NPE when generating links object with no active Stapler request
- Correctly calculate all data sizes to use 1024 bytes per increment (consistency with other parts of Jenkins) ([JENKINS-33714](#))

## 1.3 (Apr 6, 2016)

- Scalability enhancements dramatically improving UI performance with very large flow graphs ([JENKINS-33624](#))
  - Should greatly reduce CPU load on masters when viewing these graphs

## 1.2 (Apr 6, 2016)

- Fix issues introduced with Pipeline 2.0 that could produce linkage errors
- Bump to parent POM v2.5

## 1.1 (Apr 5, 2016)

- Fix log details not expanding/collapsing for pipeline steps ([JENKINS-33110](#))
- Clean up POMs and use new parent POM ([JENKINS-33188](#))
- Fix a StackOverflow Exception when navigating long pipeline flow graphs ([JENKINS-33486](#))

## 1.0 (Feb 25, 2016)

- Initial open source code release