

Deployment Notification Plugin

Plugin Information

View deployment-notification [on the plugin site](#) for more information.

This plugin integrates configuration management systems like Chef, Puppet, etc. with Jenkins by allowing Jenkins to track what files are deployed when and where by. This allows Jenkins to perform additional automated tasks upon deployment, thereby making it easier to fully automate continuous delivery pipelines/workflows.

This plugin employs an unique approach to this problem. Instead of requiring you to run these deployment tools from within Jenkins, it merely asks you to submit the report of deployment executions to Jenkins. Because of this, this plugin works with any usage model of configuration management tools; it doesn't matter if you run `puppet apply` from cron, or run `chef-client` via ssh, or anything else.

This approach hinges upon one of the key principles of continuous delivery, namely no rebuilding of binaries. Jenkins will just look at submitted reports and figure out MD5 checksums of the files deployed, and from there, figure out where it was built and tested by using its [fingerprint](#) database. For example, it can tell that the `foo.rpm` file that just got deployed on your server `bar.cloudbees.com` was originally built in `foo-build #15`, which is built from Git commit `1b75c09a` that contains Kohsuke's fix for `JENKINS-15620`.

There are so many ways to take advantages of such wealth of information, and this plugin offers several of those out of the box:

- Adds a new trigger to Jenkins that lets you say "start building this job when artifact(s) from `foo-build` are deployed on N servers". When a build is triggered in this manner, it receives parameters from the correct build of `foo-build`, allowing you to continue a pipeline.
- Adds a new promotion criteria for [promoted builds plugin](#). In this way, builds of `foo-build` whose artifacts hit production gets a star in the build history, making it easy to see how changes flow in your deployment pipeline.

I'm looking for feedback on other useful ways to expose this information, so please [file your ideas as tickets](#). The plugin also offers an internal extension point (`DeploymentListener`) so that other plugins can hook into this information.

This plugin defines all the models, listeners, conditions, as well as integration with other plugins, but it does not understand tool-specific execution report formats. Therefore, users would also have to install a plugin specific to the configuration management tool you use, such as [Puppet plugin](#) and [Chef Tracking Plugin](#).

Developing integration with another configuration management tool

[Puppet plugin](#) is the reference implementation that shows you how to build on top of this plugin. It'd be a good starting point.

Workflow integration

Since v1.3 this plugin is not only integrated with Workflow plugin but it also adds a Workflow step called [awaitDeployment](#). The example below shows you how to use the [awaitDeployment](#) step on a Workflow DSL.

```
node('unix-slave') {
  git changelog: false, credentialsId: '', url: 'git@github.com:fbelzunc/spring-petclinic-clickstart.git'
  sh 'mvn clean package -Dmaven.test.skip=true'

  step([$class: 'ArtifactArchiver', artifacts: 'target/petclinic.war', fingerprint: true])
}

awaitDeployment env: 'production', threshold: 1
```

Changelog

Version 1.4 (April 3, 2016)

- Fix [JENKINS-33988](#) Inject deployment env variables in the build

Version 1.3 (Jun 27, 2016)

- Fix [JENKINS-28632](#) Workflow support for Deployment Notification trigger
- Fix [JENKINS-23569](#) Missing index.jelly resource for `org.jenkinsci.plugins.deployment.promoted_builds.DeploymentPromotionCondition`

Version 1.2 (Jun 16, 2014)

- Fixed a bug that a trigger didn't restore the state correctly when Jenkins starts up.

Version 1.1 (Mar 31, 2014)

- Components requiring promoted-builds plugin weren't marked as optional.

Version 1.0 (Feb 17, 2014)

- Initial release