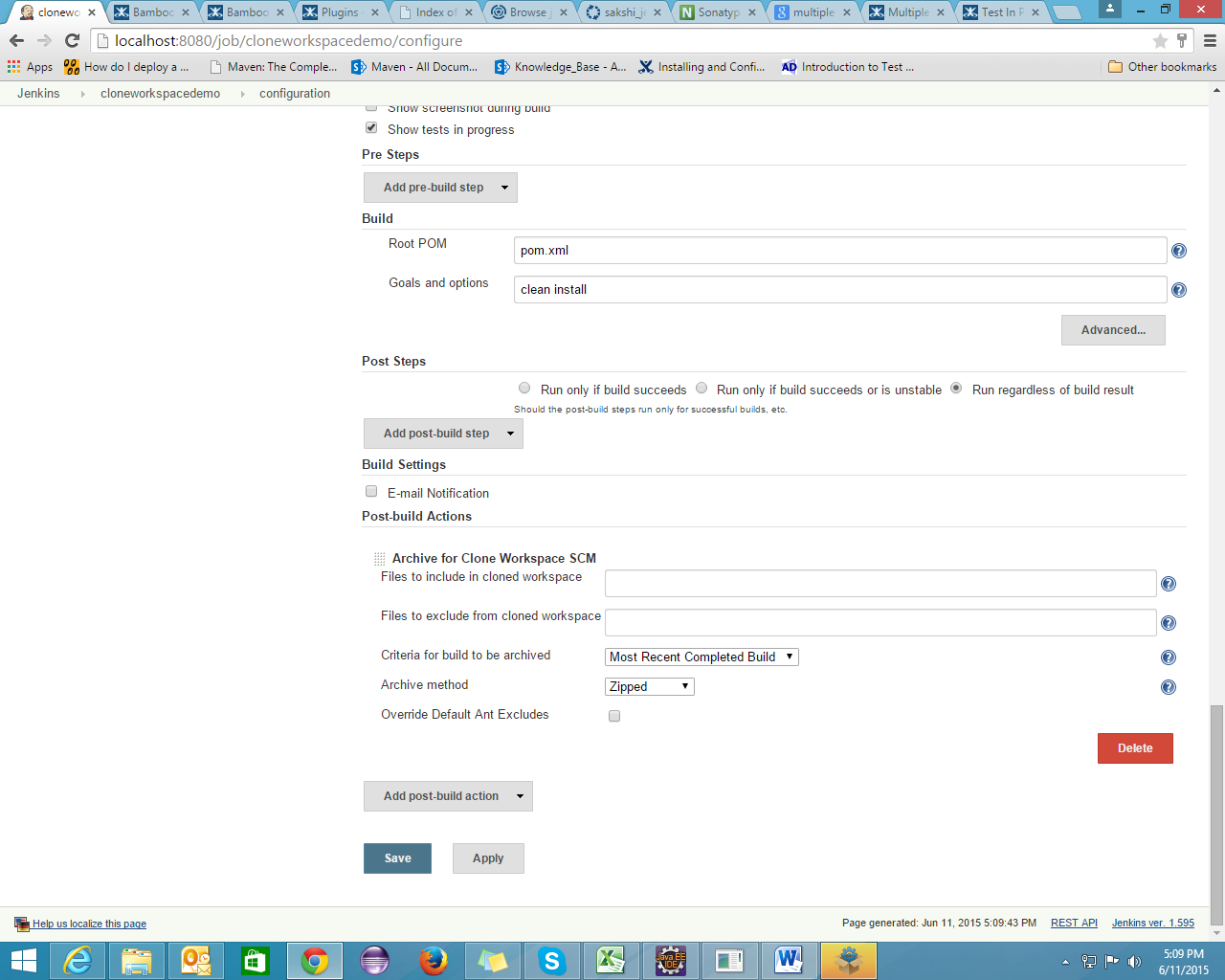
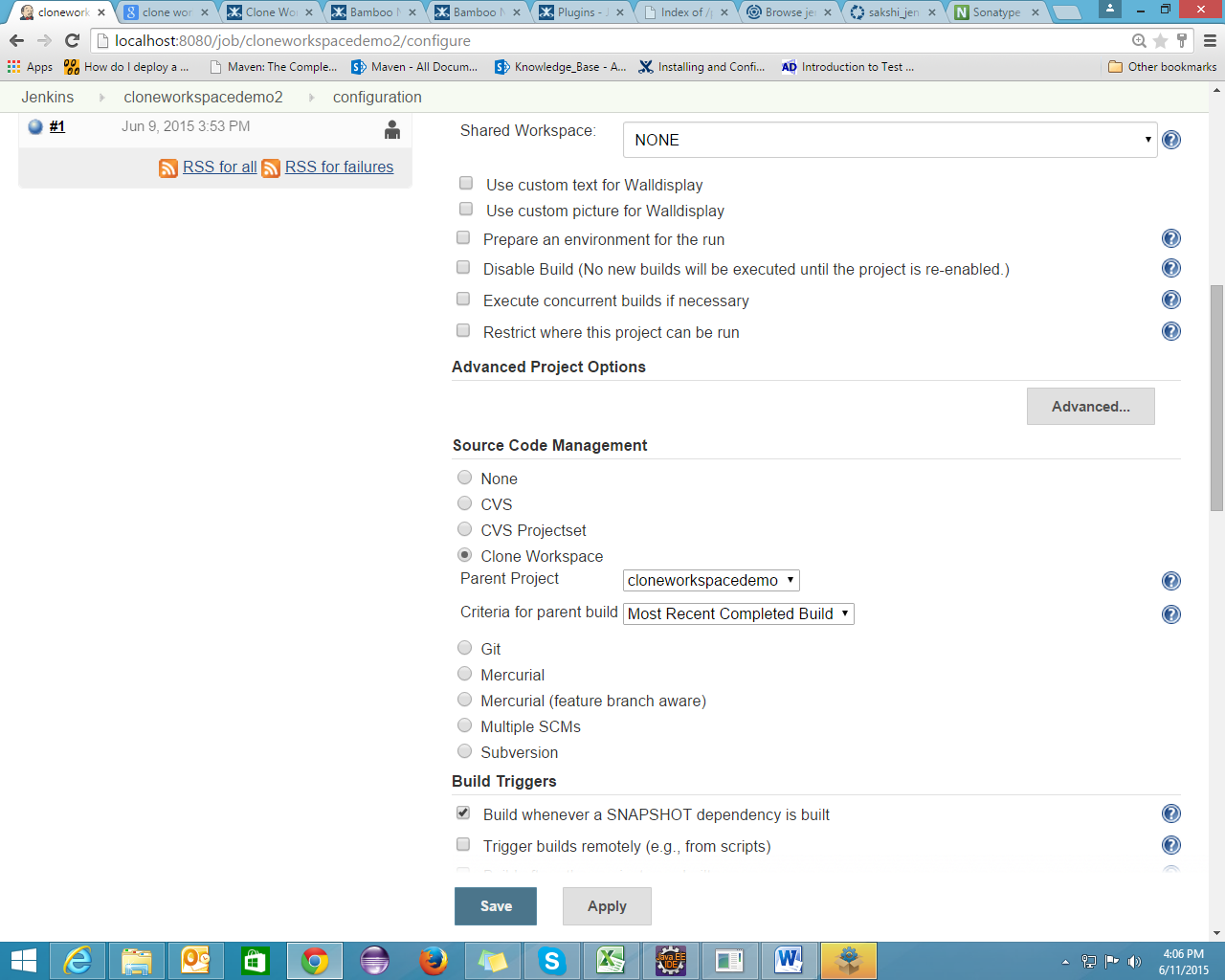
**Clone Workspace Plugin**

This plugin makes it possible to archive the workspace from builds of one project and reuse them as the SCM source for another project.

**Configurations:**

**Global Level Config:**





This job will use the workspace of its parent project.

**Shared Workspace Plugin**

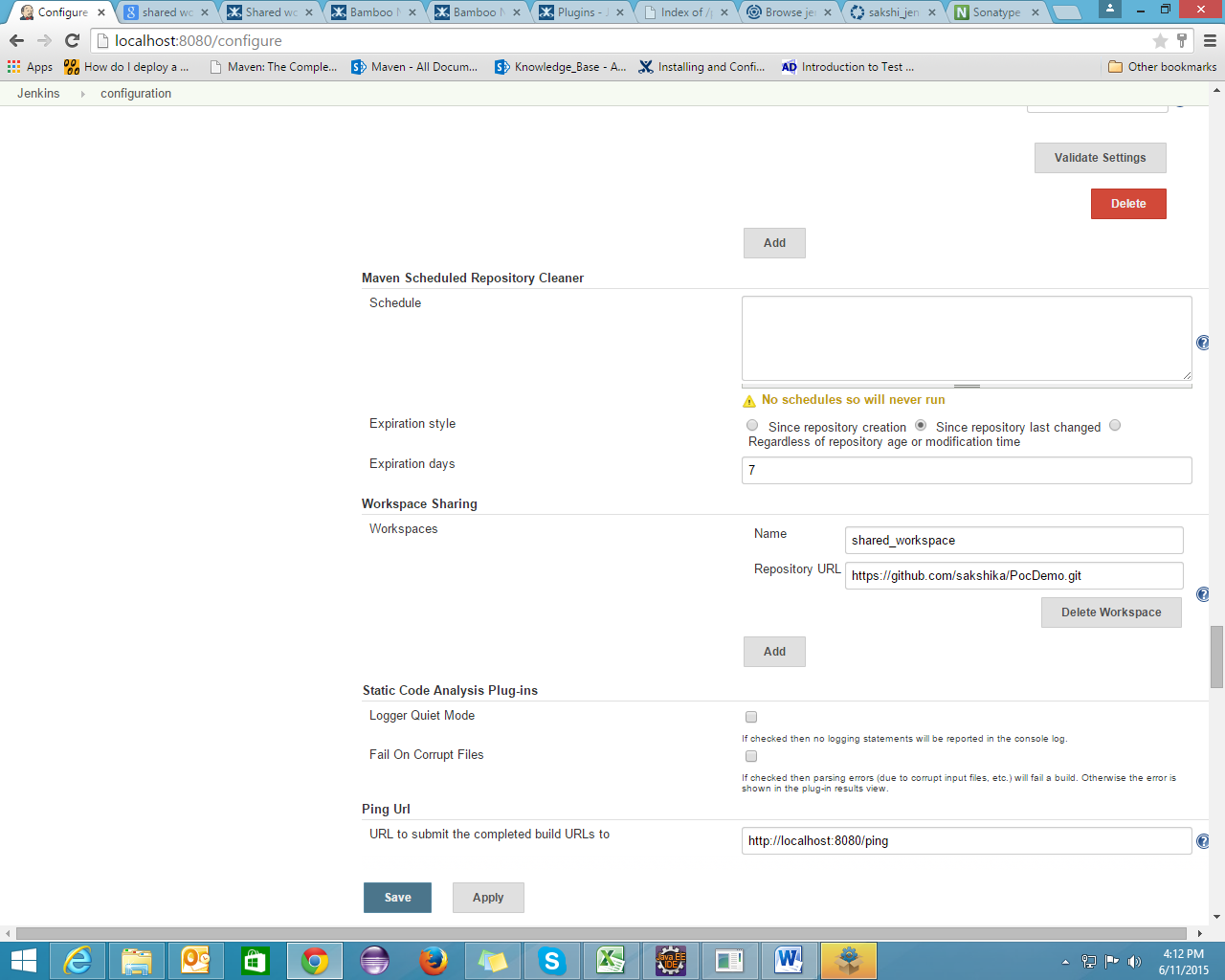
This plugin allows to share workspaces by Jenkins jobs with the same SCM repos.

It should save you some disk space, if you have different jobs with identical repos.

**Configurations:**

Go to Manage Jenkins -> Configure System -> Workspace Sharing

Specify the name and repository URL. Your shared workspace will get created.

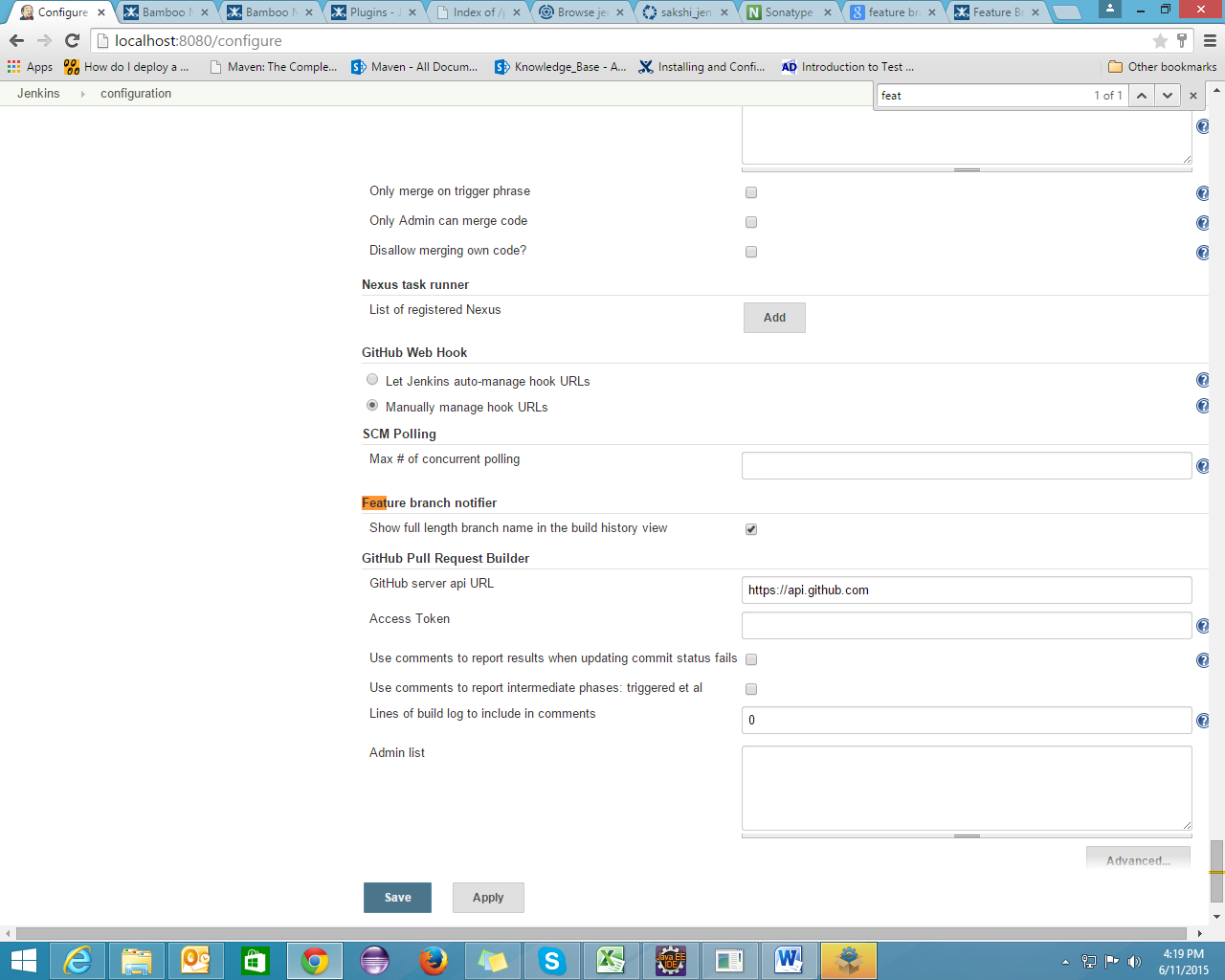


Feature Branch Notifier Plugin

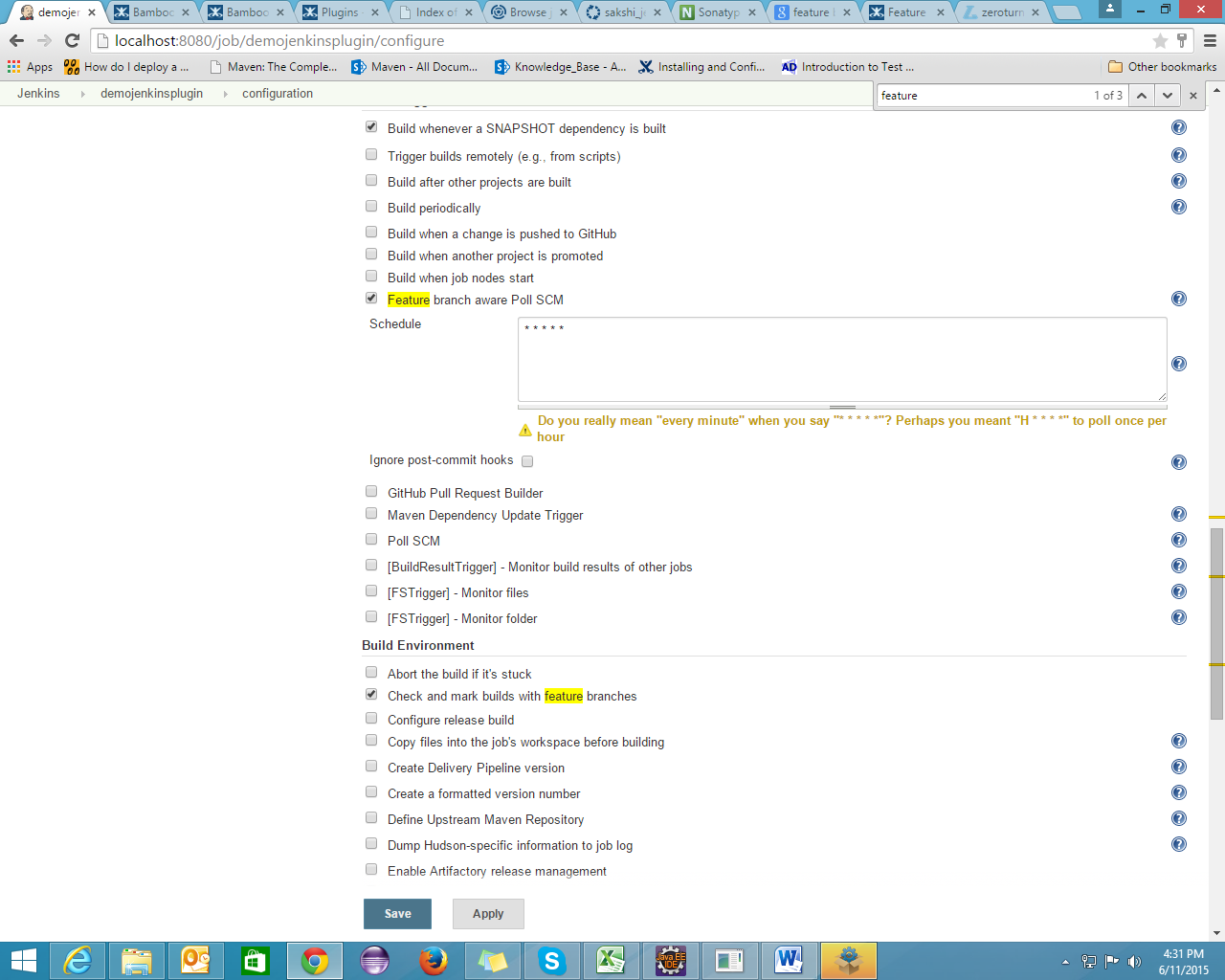
If there is any change in the feature branch, jenkin build will get trigger.

**Configurations:**

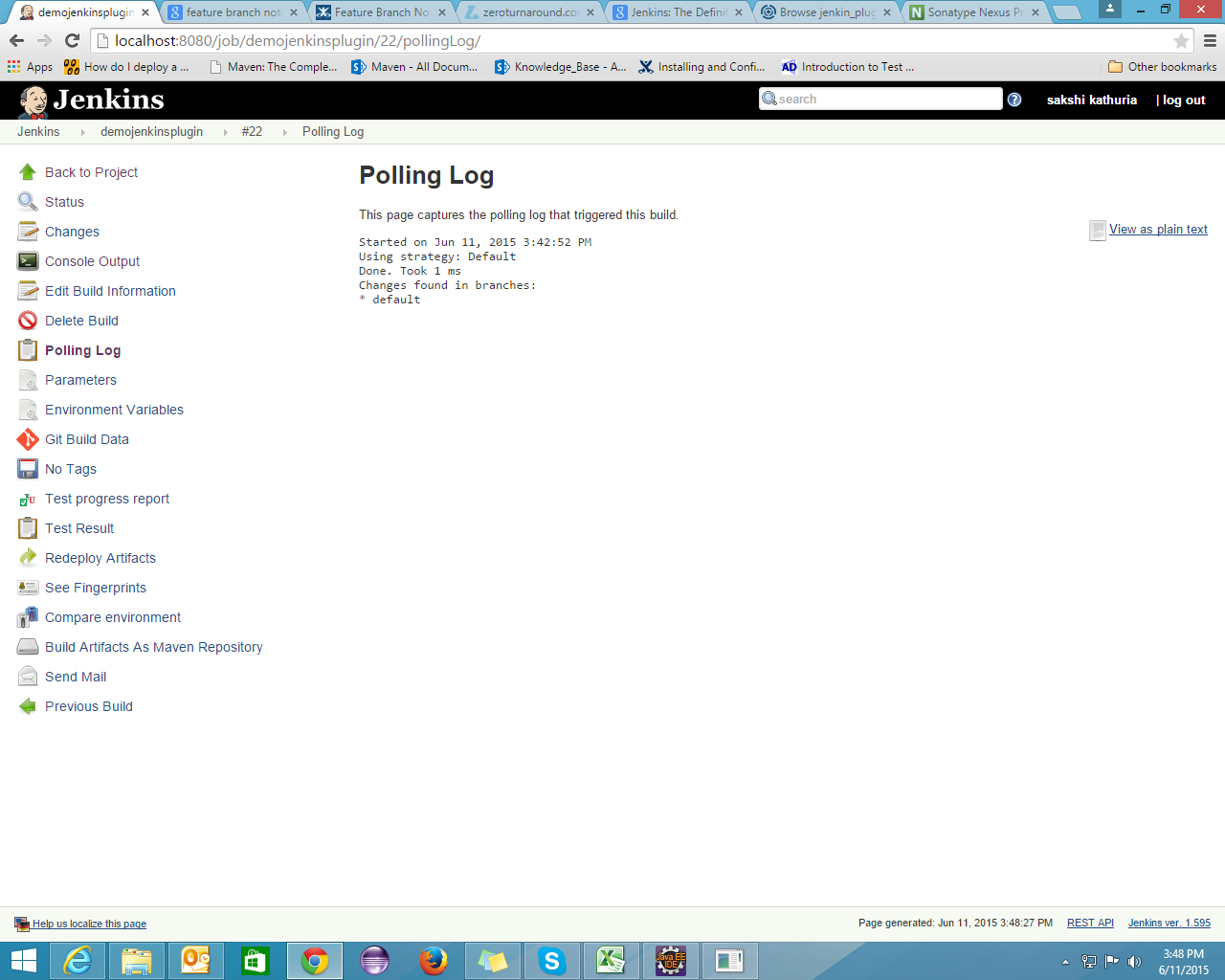
**Global Config:**



**Job Level Config:**



Build will automatically get trigger.

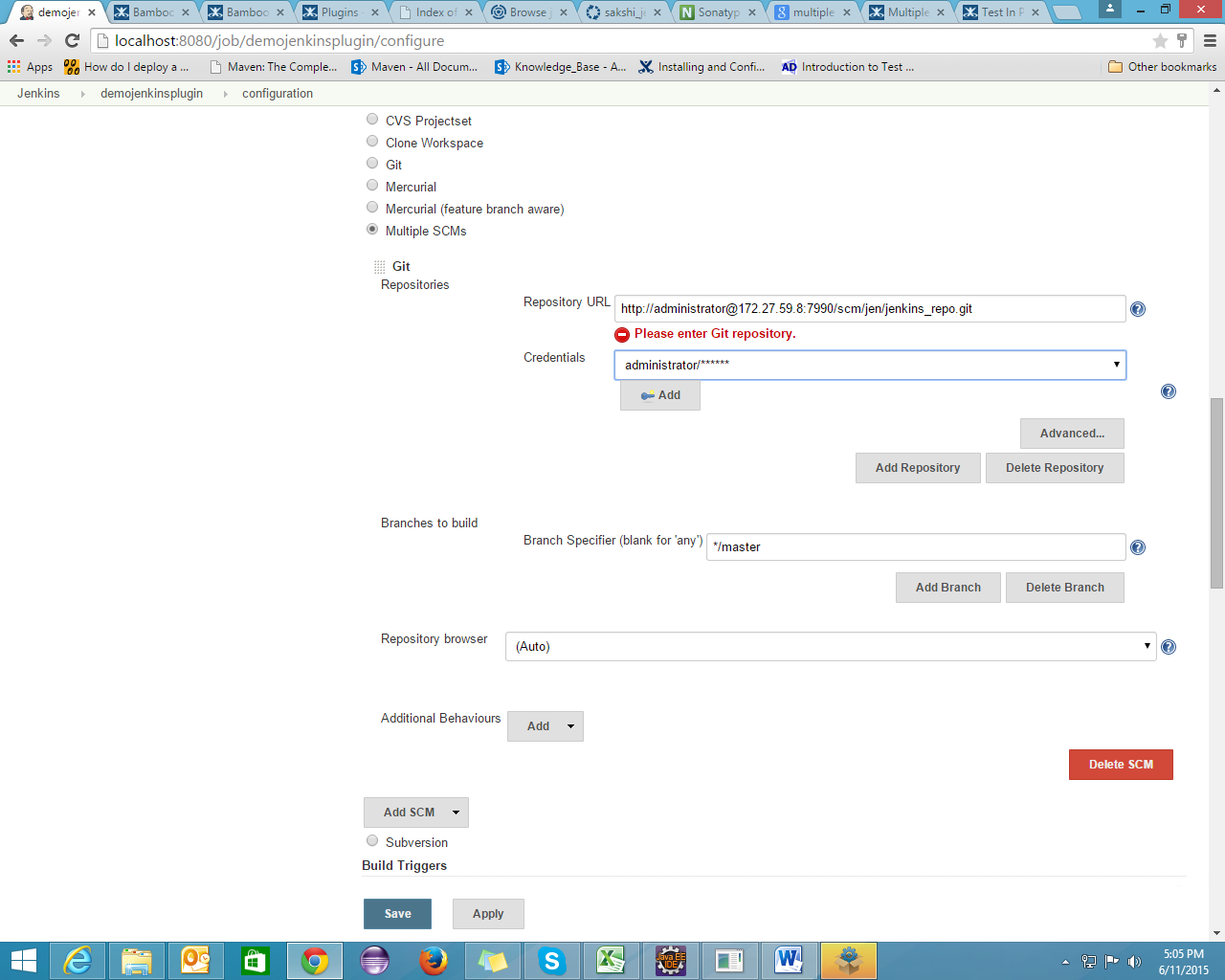


**Multiple SCMs Plugin**

It allows a job to check out sources from multiple SCM providers.

In the SCM section of the Jenkins job configuration screen, choose 'Multiple SCMs'. You'll then see a drop-down list of the available SCM plugins which can be configured similar to the way build steps are configured for a freestyle job. When chosen, each SCM plugin will present its normal set of configuration options.

**Configurations:**

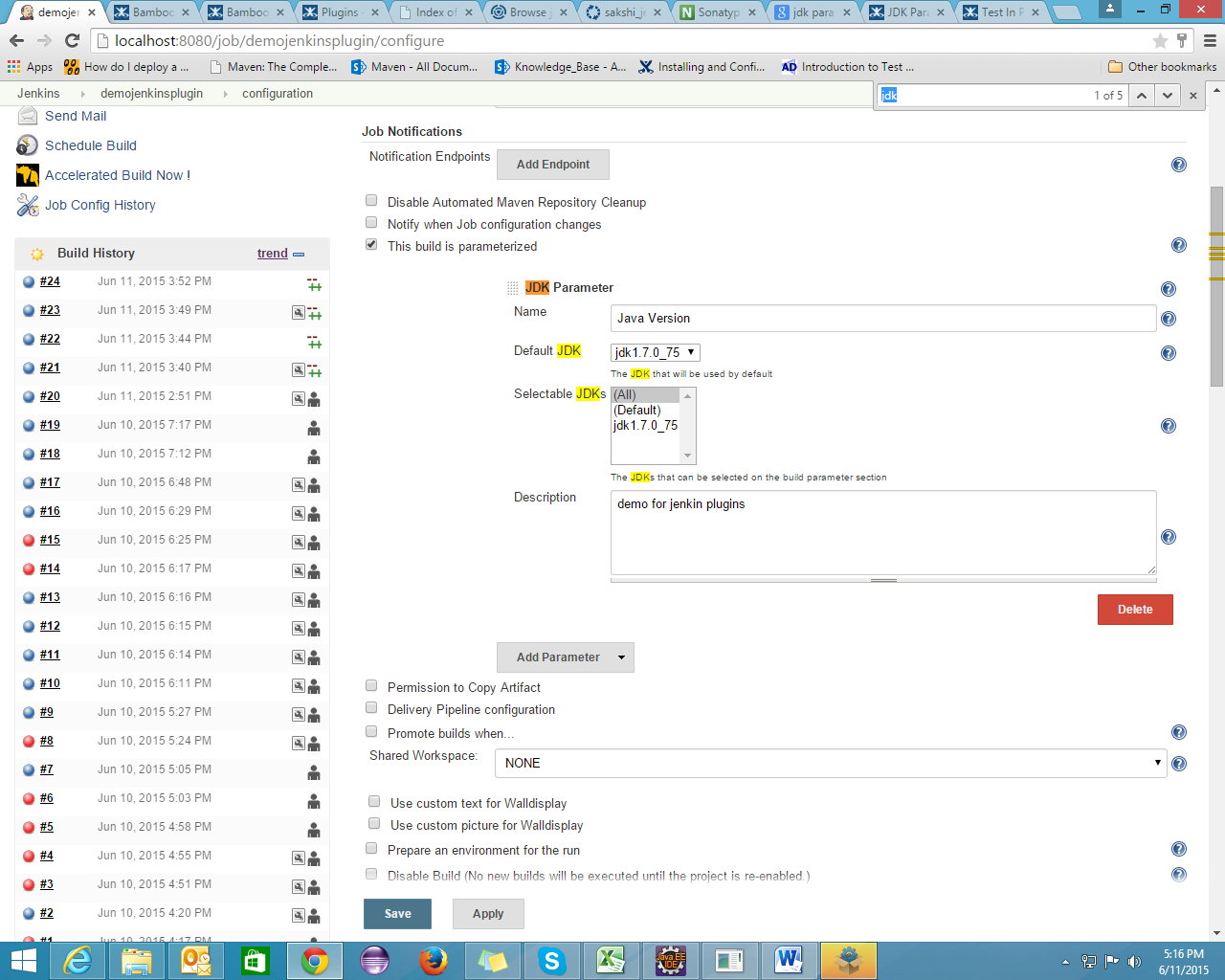


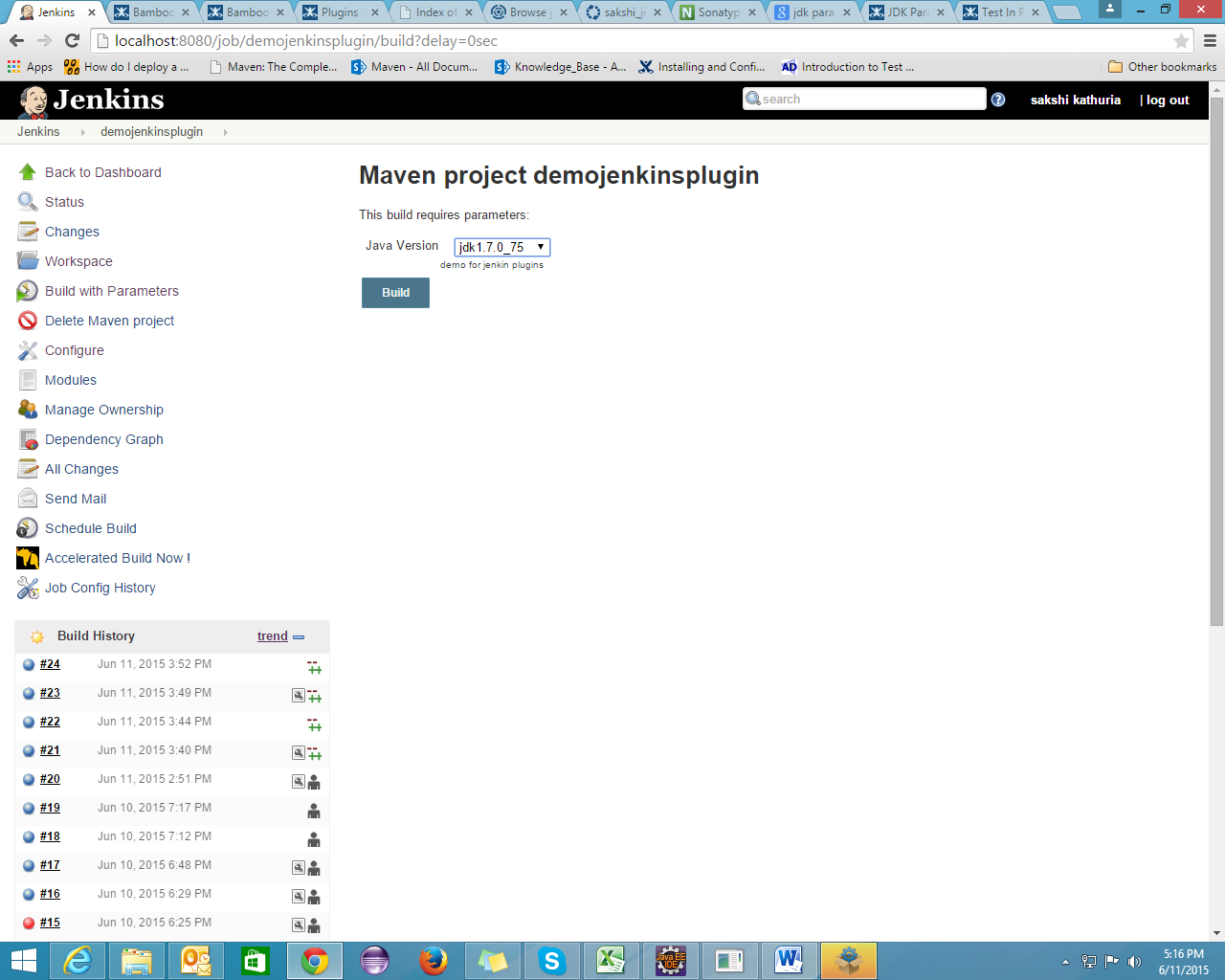
**JDK Parameter Plugin**

This plugin adds a build parameter which can be used to set the JDK to be used with a job on a per-build basis. The JDKs that can be chosen from can be specified on the configuration page of any job that has the JDK parameter enabled.

**Configurations:**

The plugins configuration is located in the 'Parameters' section of the job configuration UI.   
There will be a dropdown list to select what should be used as the default JDK on the build parameters page and a multi-selection box where you can choose what JDKs are available to pick from as a build parameter for that job.

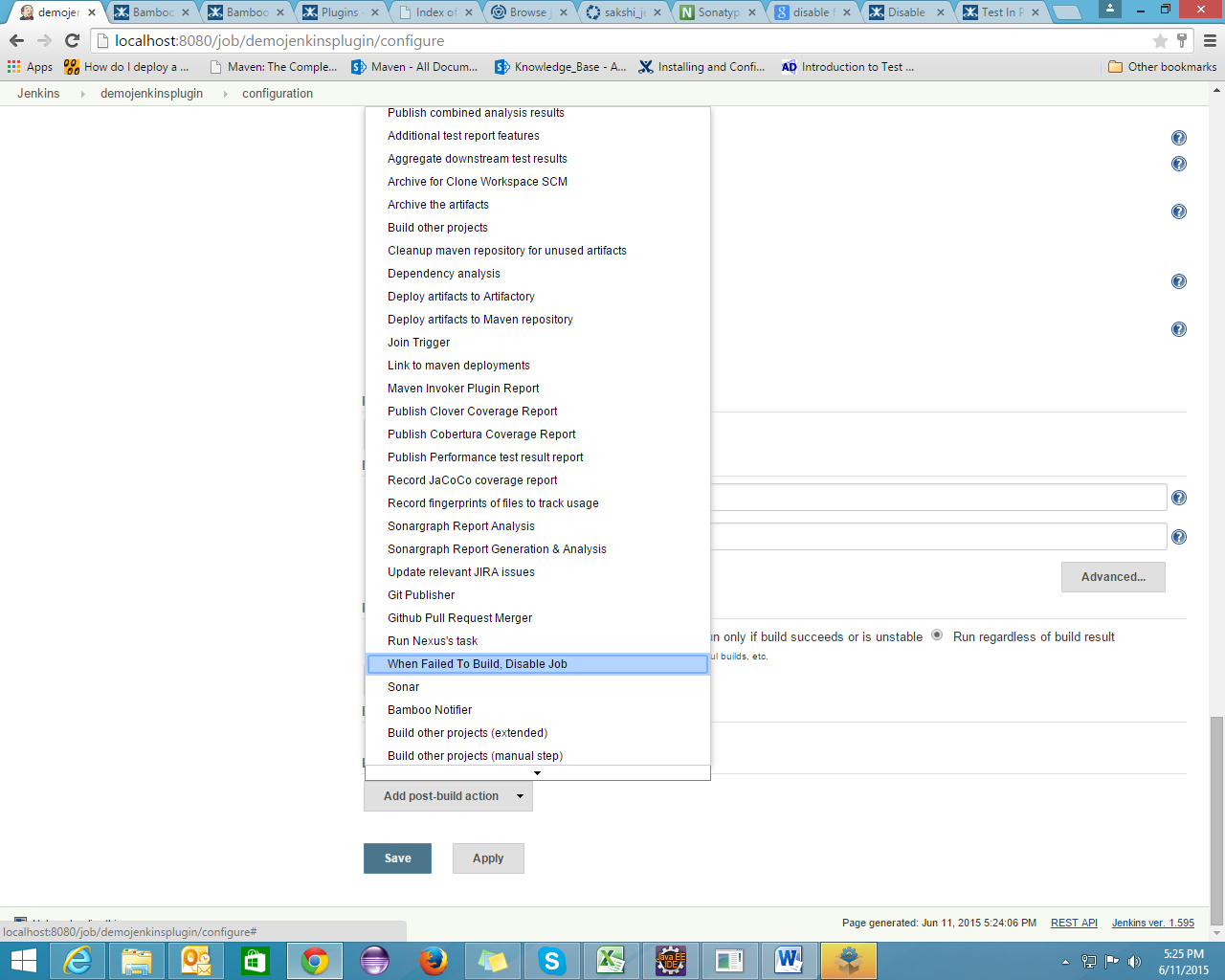


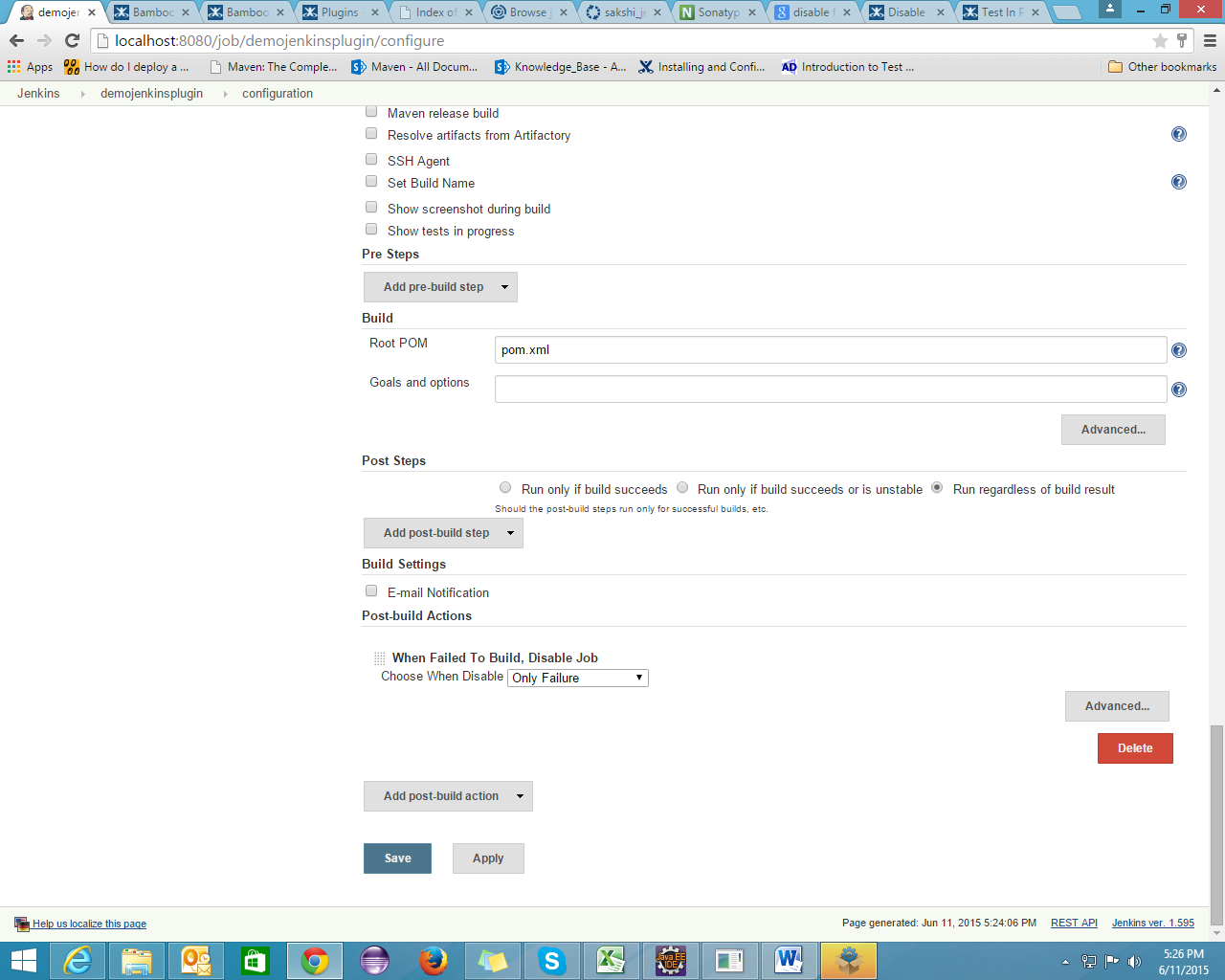


**Disable Failed Jobs**

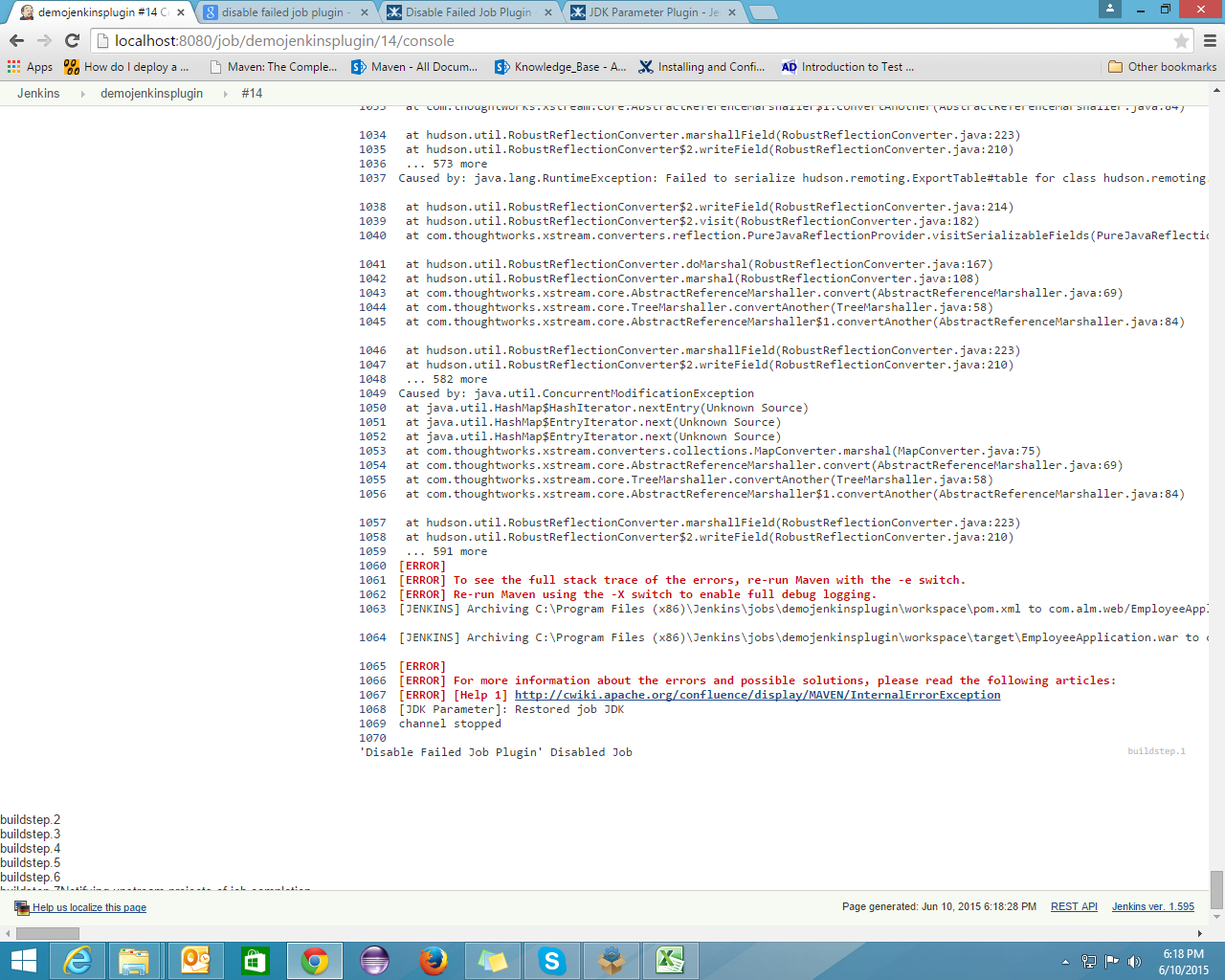
This Plugin disables a Job when a Build fails.

**Configurations:**

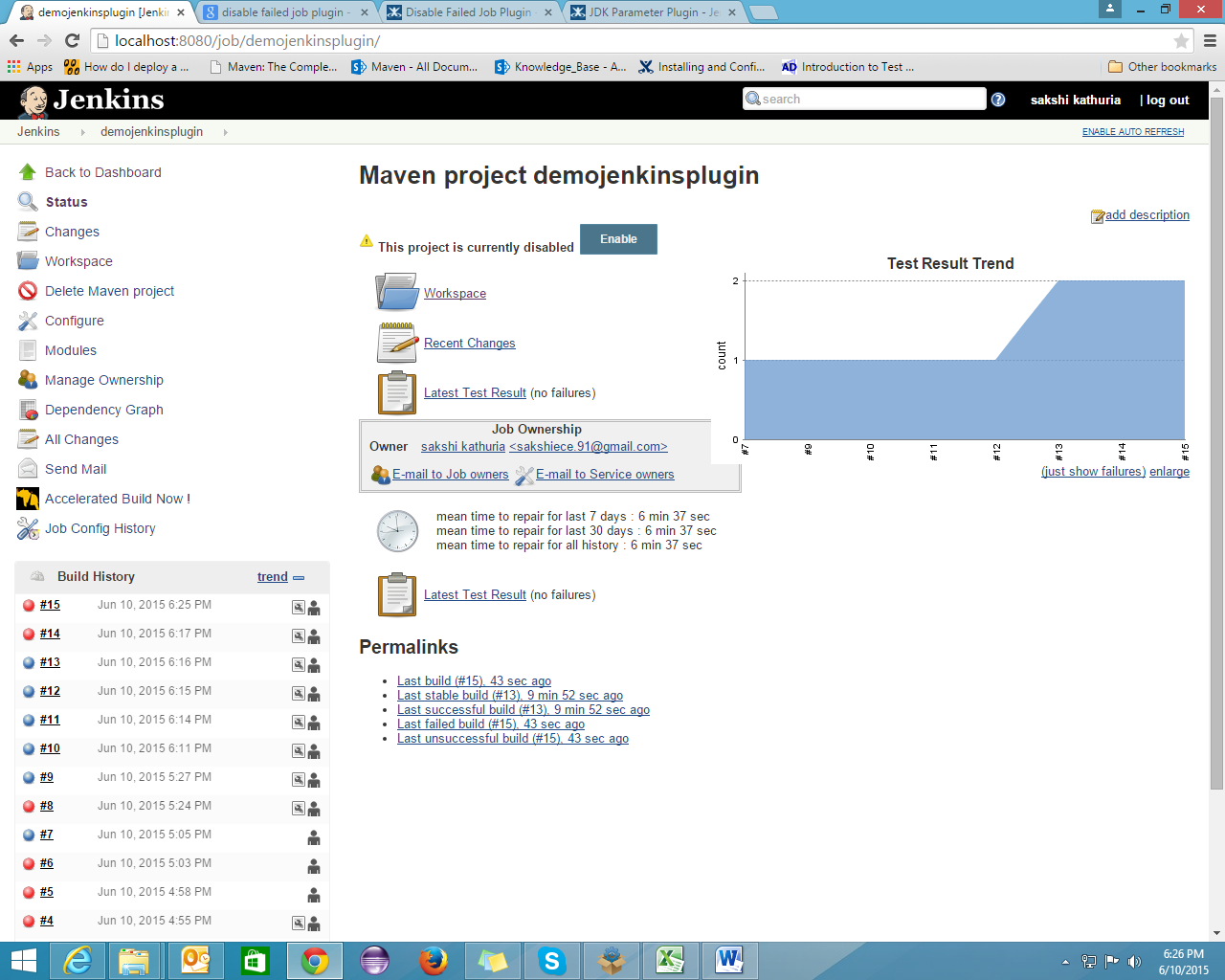




Build Failure:



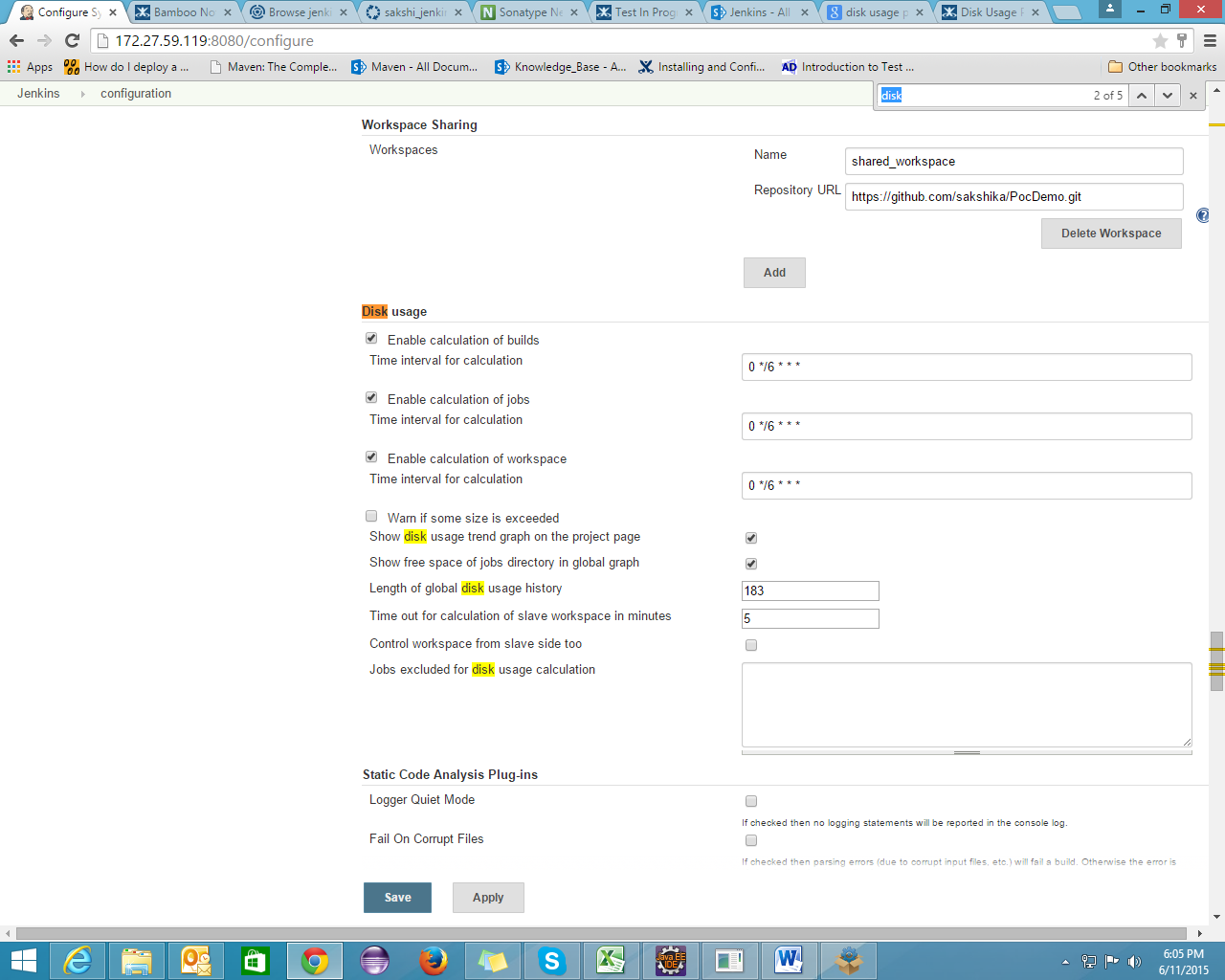
When the build got failed, job will get disable. We have to manually enable it as shown below



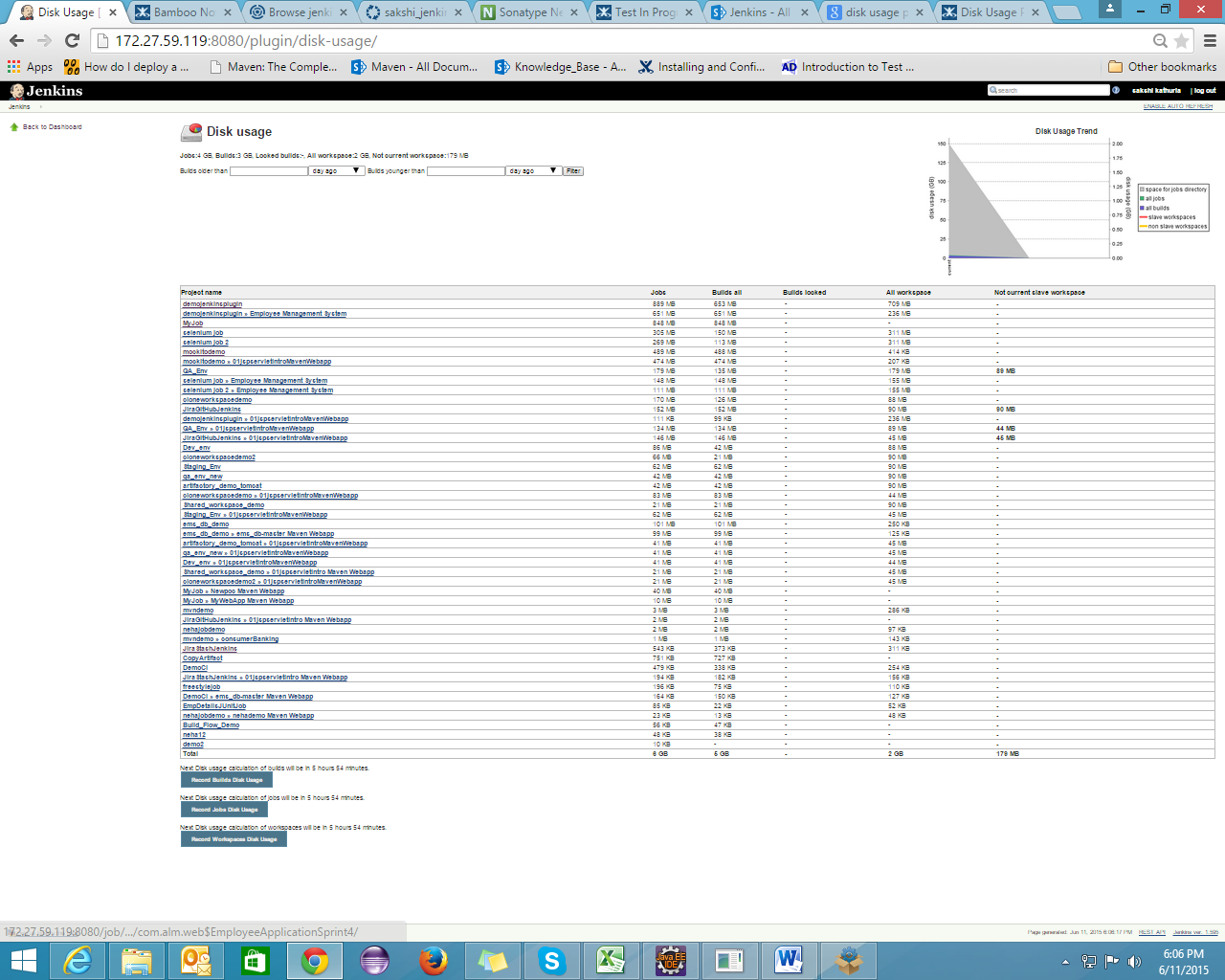
**Disk Usage Plugin**

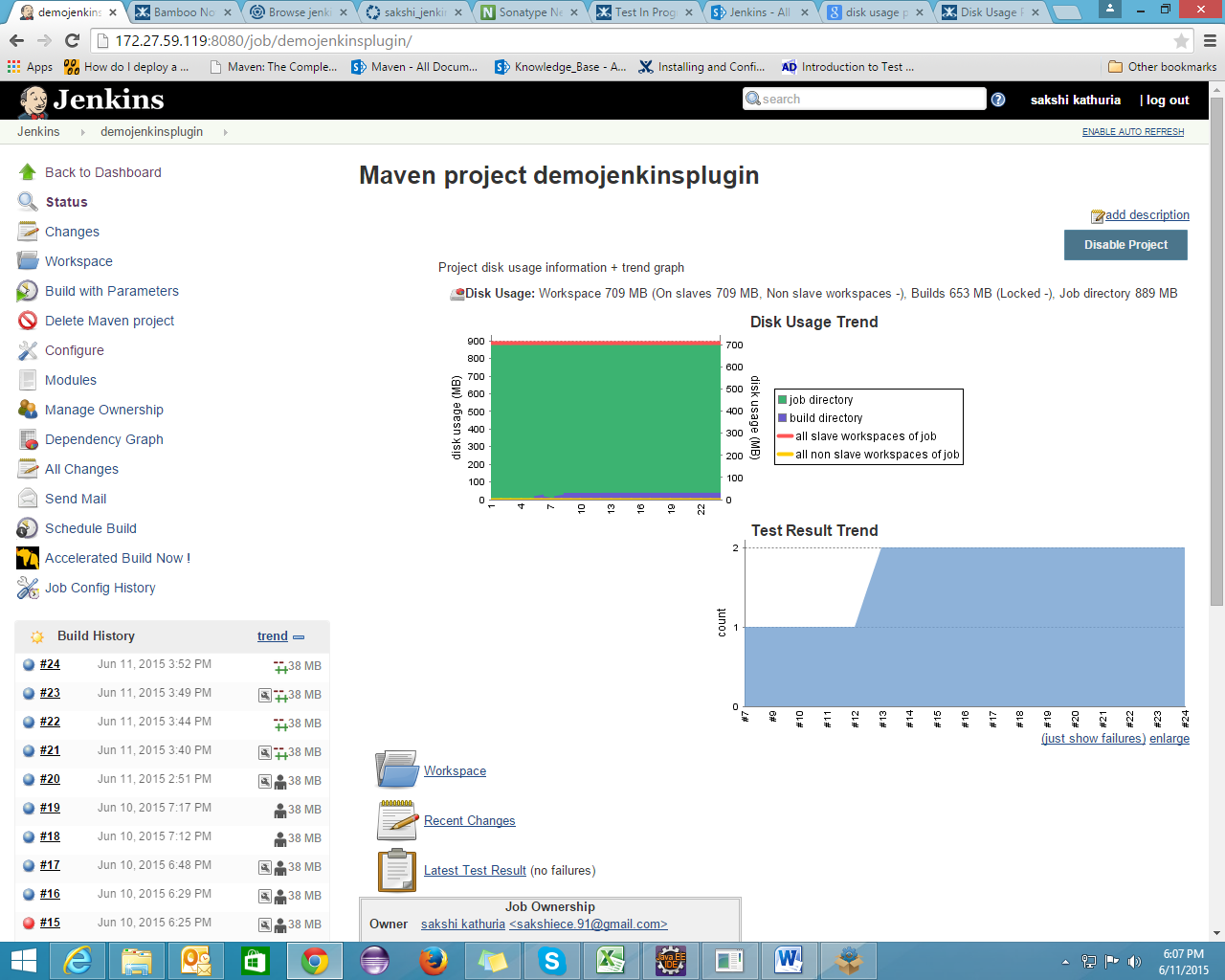
Will show you how much disk space is covered by yours jobs.

**Configurations:**



Shows the disk space covered by all the jobs:



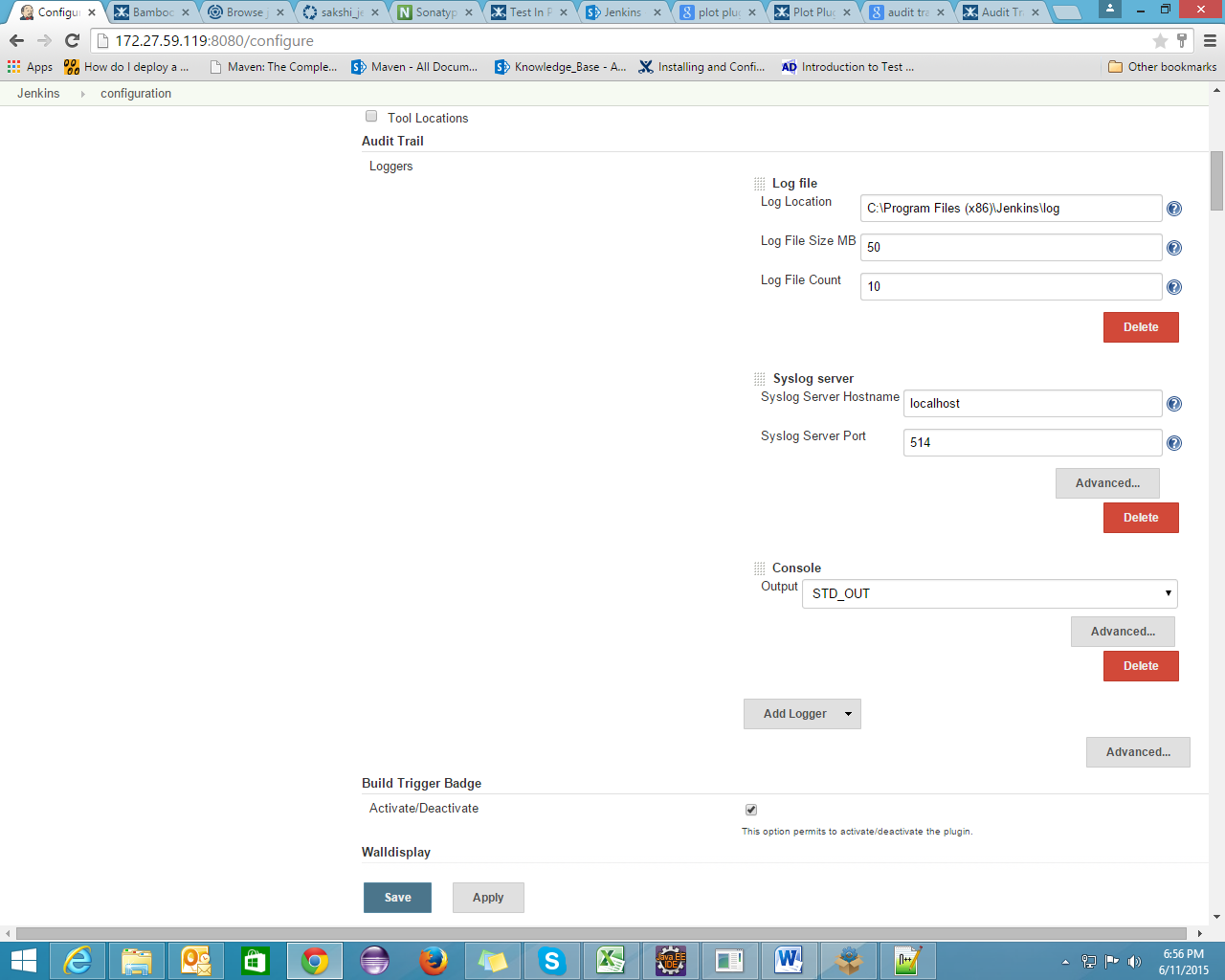


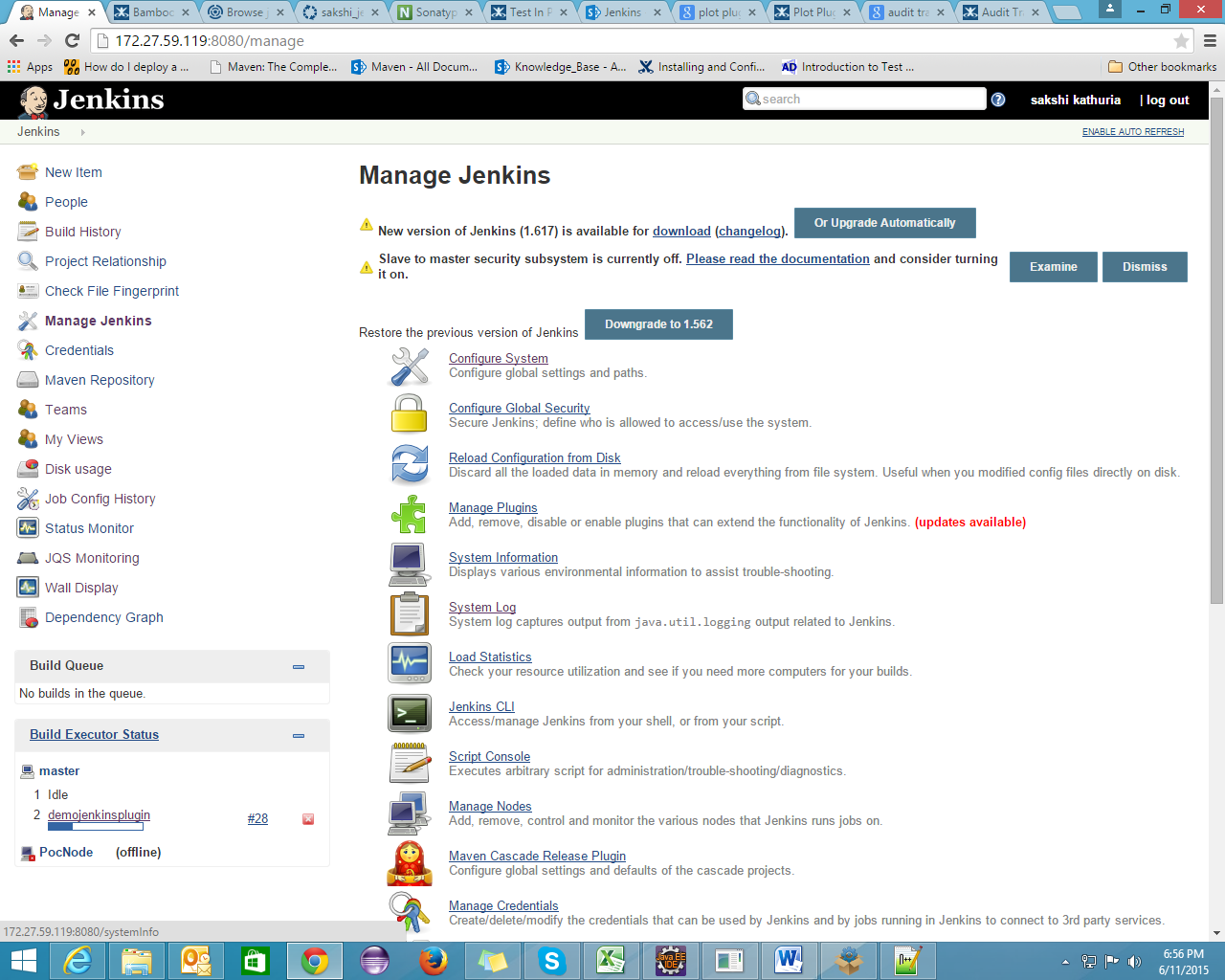
**Audit Trail Plugin**

Keep a log of who performed particular Jenkins operations, such as configuring jobs.  
This plugin adds an Audit Trail section in the main Jenkins configuration page. Here you can configure log location and settings (file size and number of rotating log files), and a URI pattern for requests to be logged.

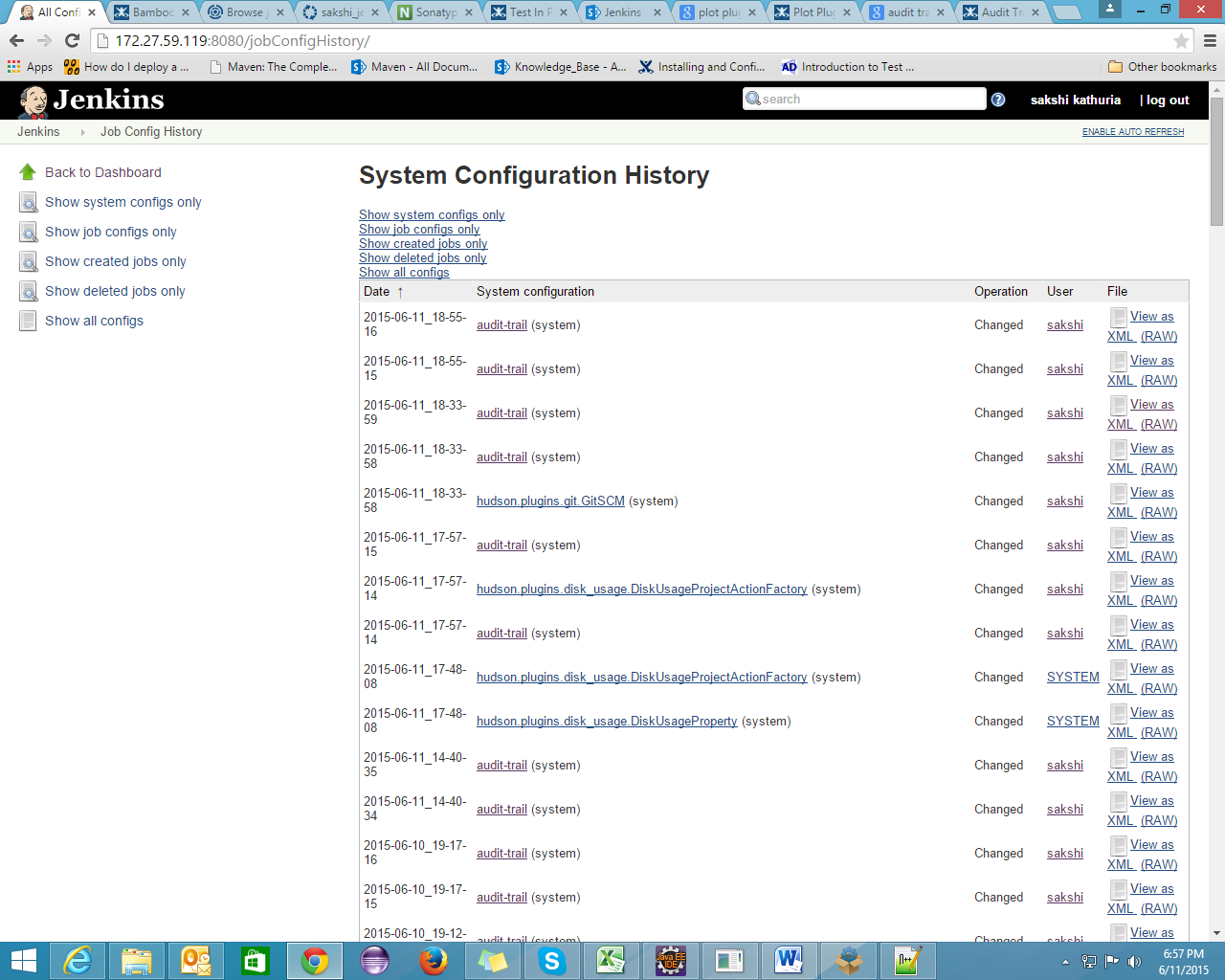
The log is written to disk as configured and recent entries can also be viewed in the Manage / System Log section.  
Also see the [JobConfigHistory Plugin](https://wiki.jenkins-ci.org/display/JENKINS/JobConfigHistory+Plugin) for recording actual changes made to job configurations.

**Global Settings:**





**System Configuration History:**

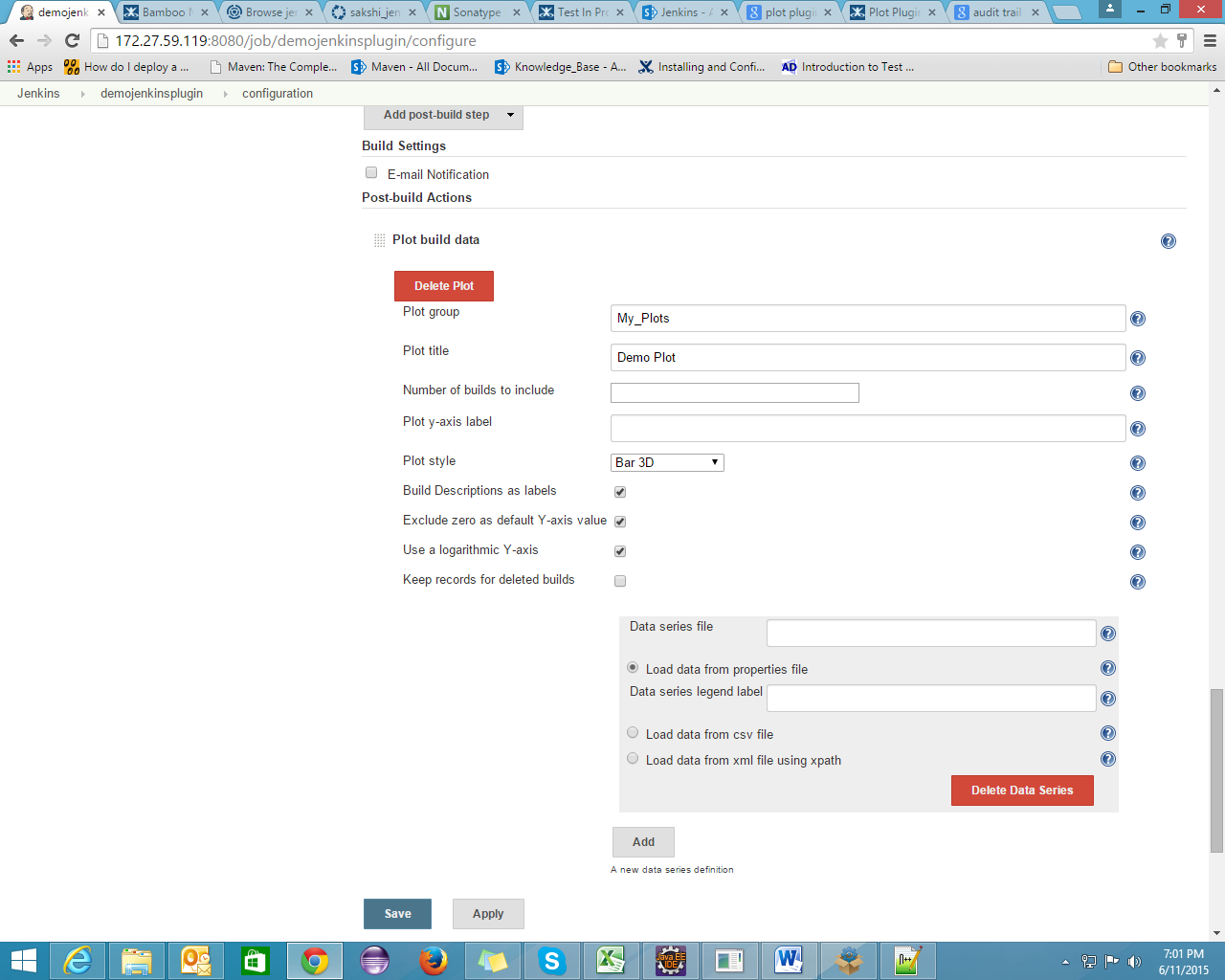


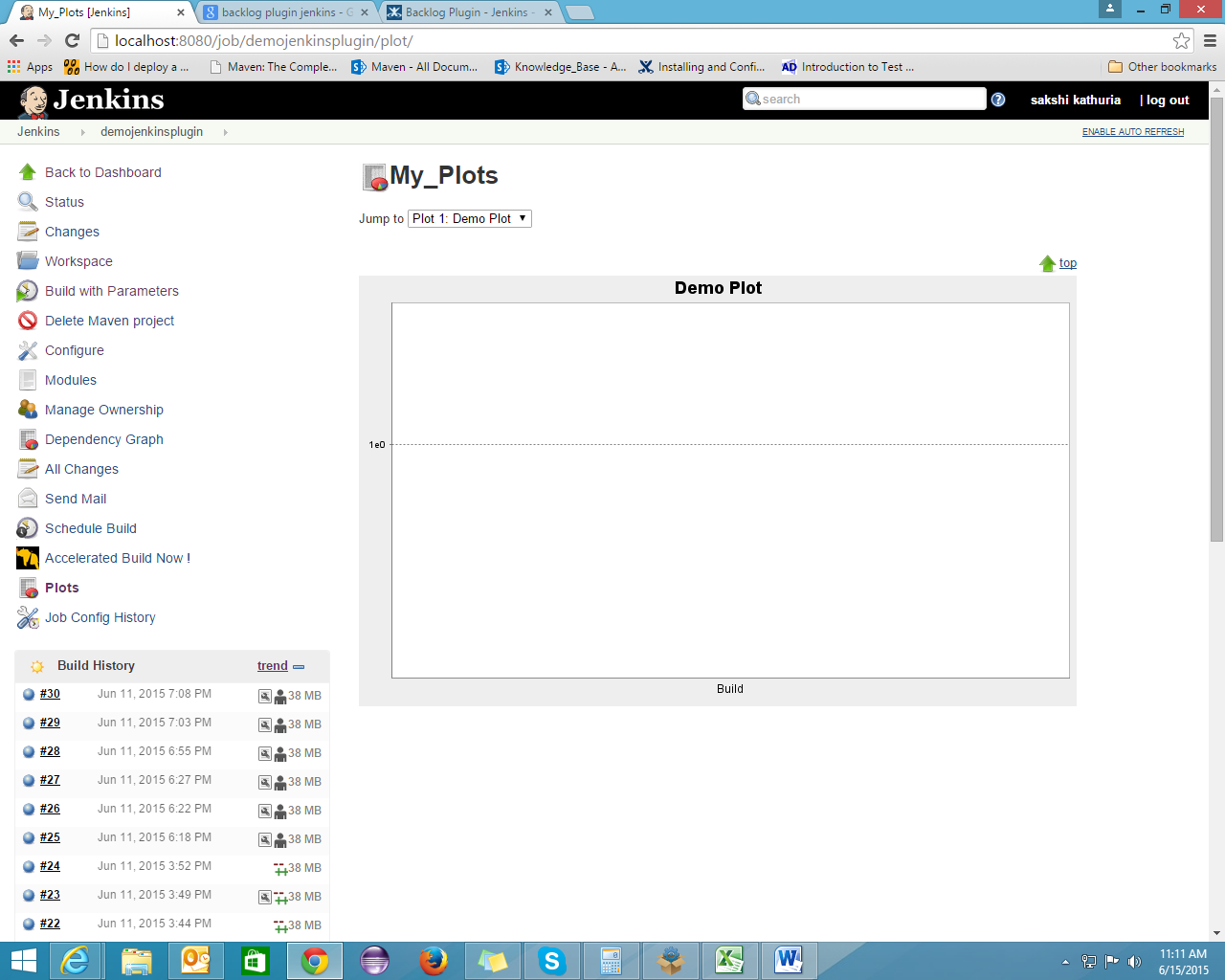
**Plot Plugin**

This plugin provides generic plotting (or graphing) capabilities in Jenkins.

This plugin will **plot** one or more **single values variations across builds** in one or more plots. Plots for a particular job (or project) are configured in the job configuration screen, where each field has additional **help information**. Each plot can have one or more lines (called data series). After each build completes the plots' data series latest values are pulled from **Java properties file(s), CSV file(s), or XML file(s)** via an XPath (which you should have generated during the build) somewhere below you workspace. Data for each plot is stored in a CSV file within the job's root project directory.

It can generate various kind of plots, including **Area**, **Bar**, **Line**, **Stacked Bar**, **Waterfall**...

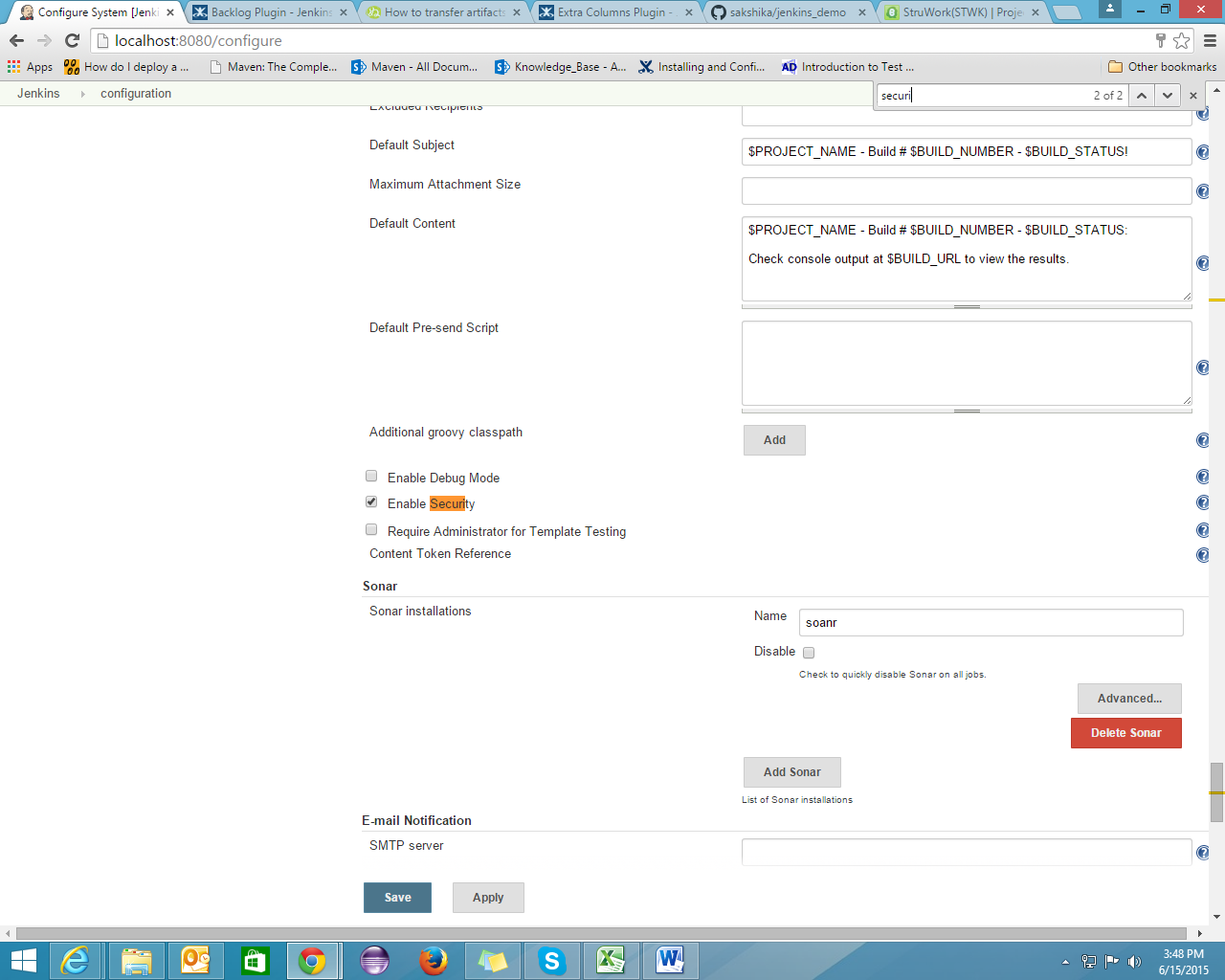




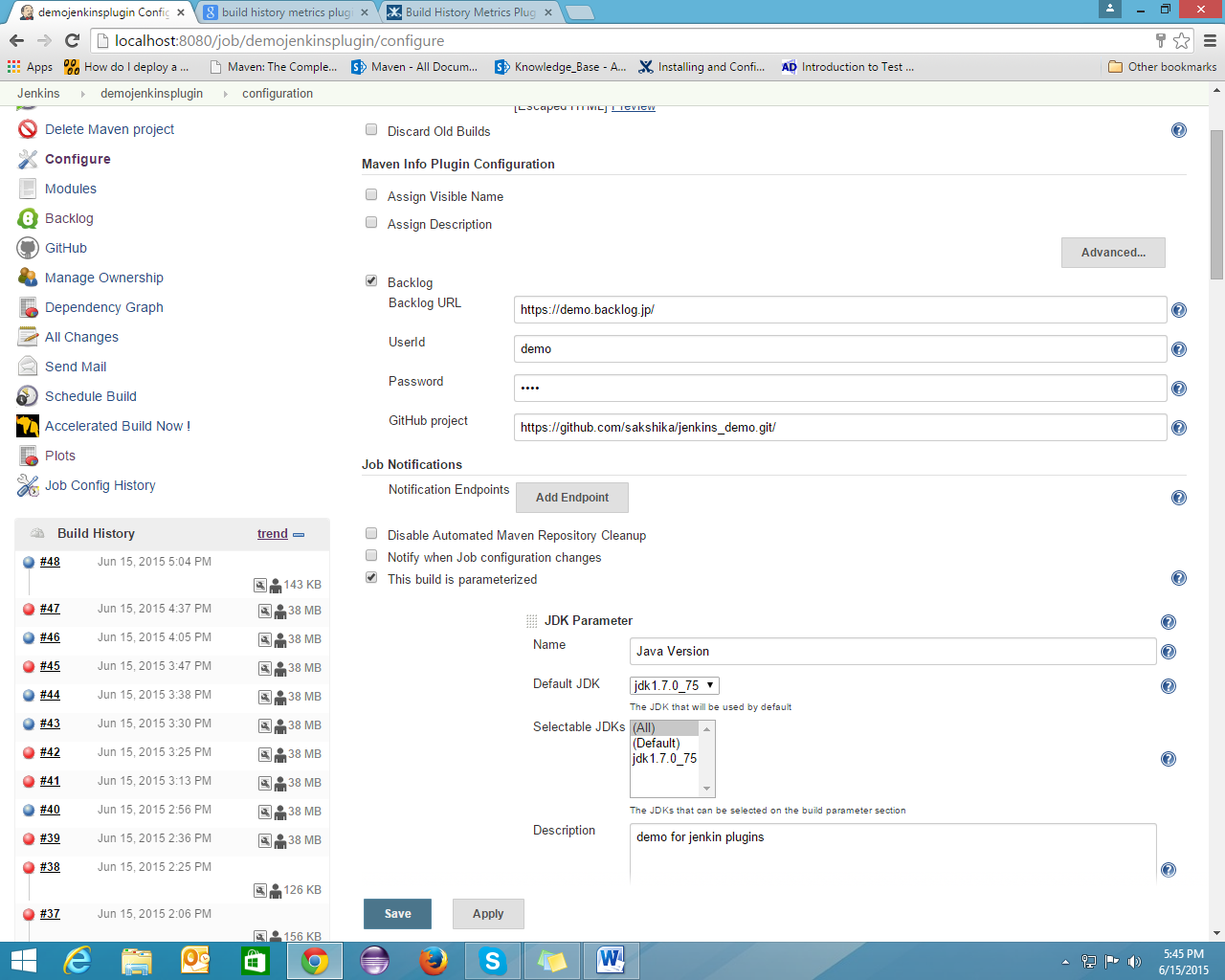
**Backlog Plugin**

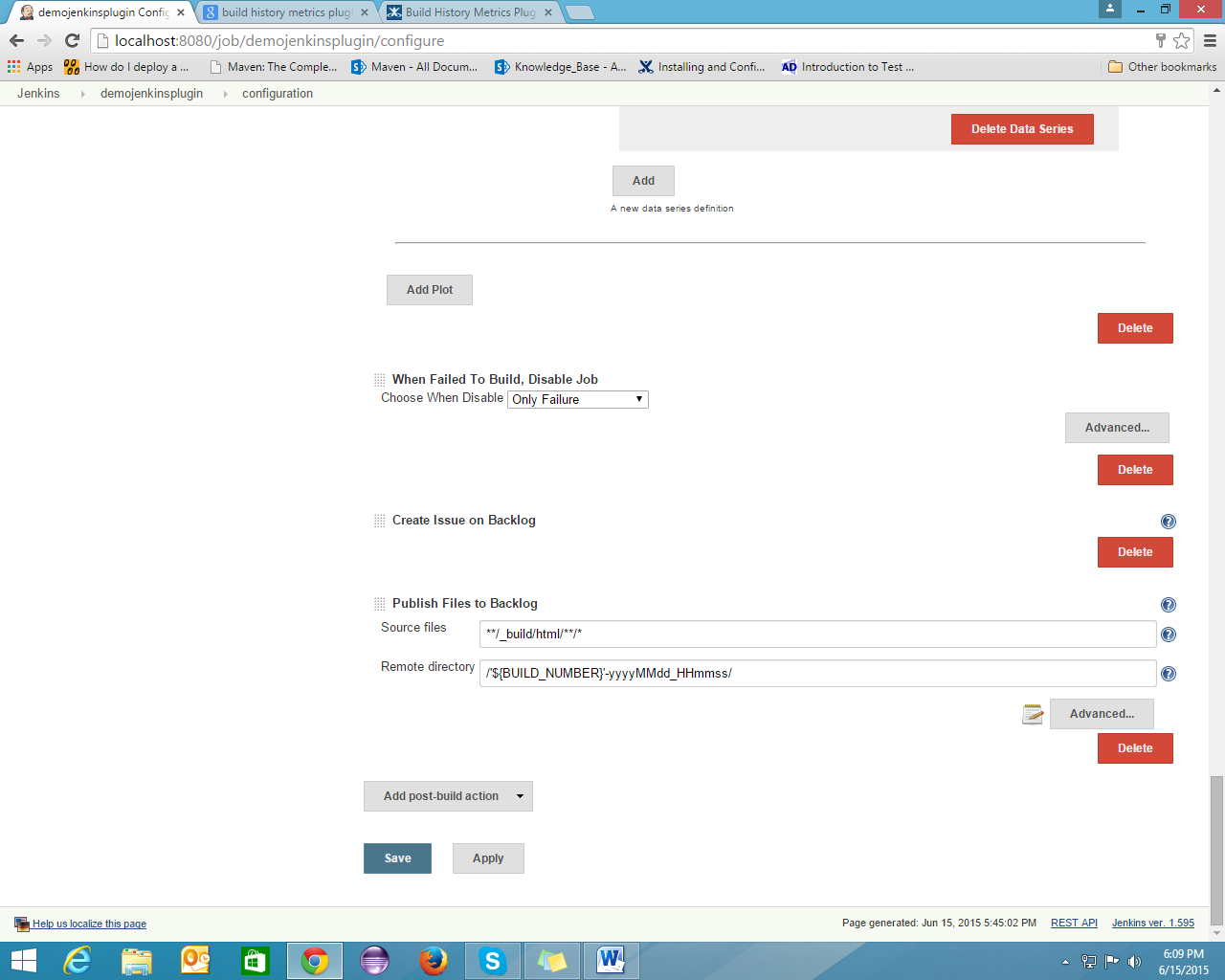
This plugin integrates [Backlog](http://backlogtool.com/) ([for Japanese users](http://backlog.jp/)) to Jenkins. Backlog is an issue tracking system provided by [Nulab](http://nulab-inc.com/).

**Global Configuration:**



**Job Level Configuration:**





**Facing issue**: Publisher hudson.plugins.backlog.BacklogWebdavPublisher aborted due to exception

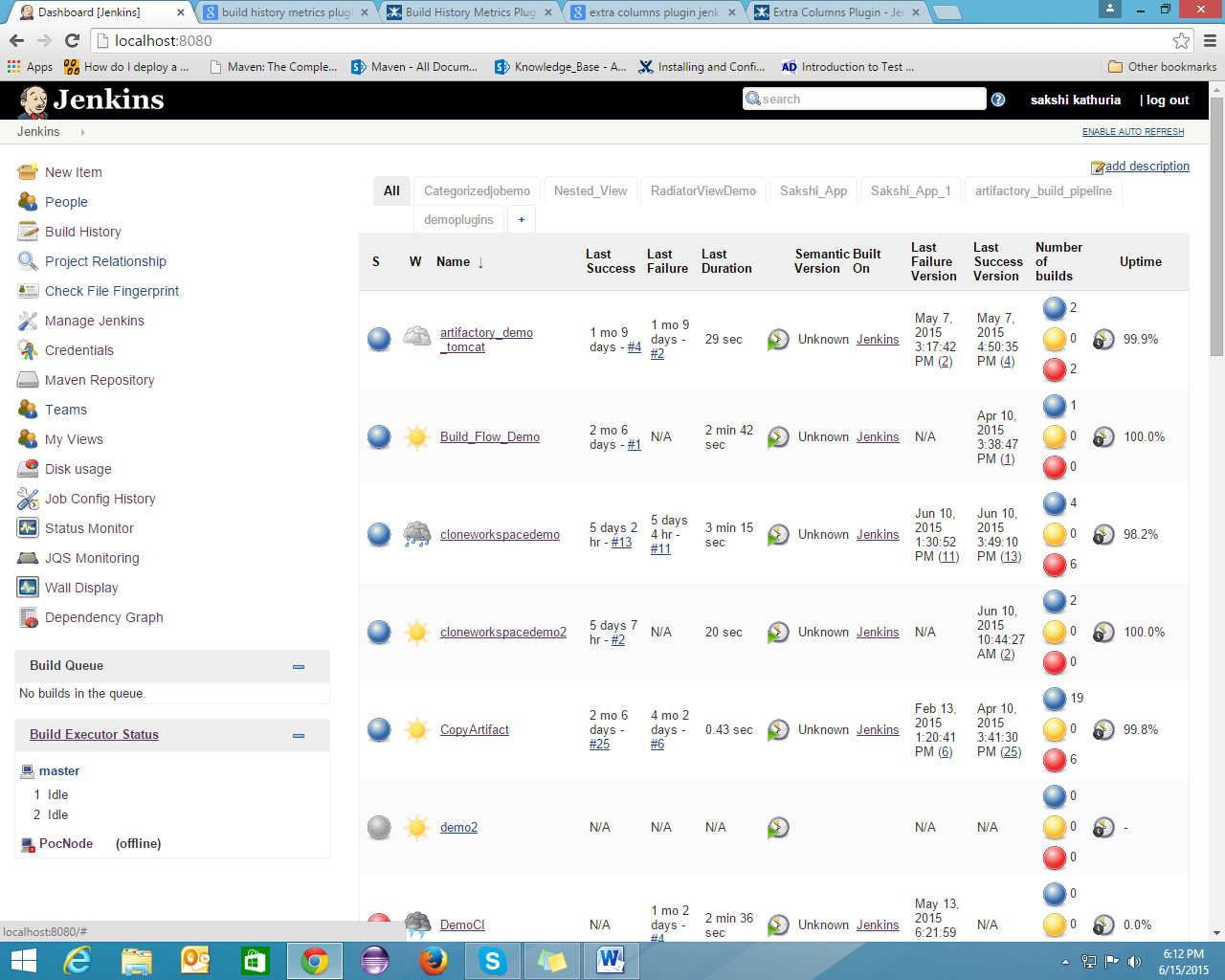
[java.lang.IllegalArgumentException](http://stacktrace.jenkins-ci.org/search?query=java.lang.IllegalArgumentException): Failed login to Backlog

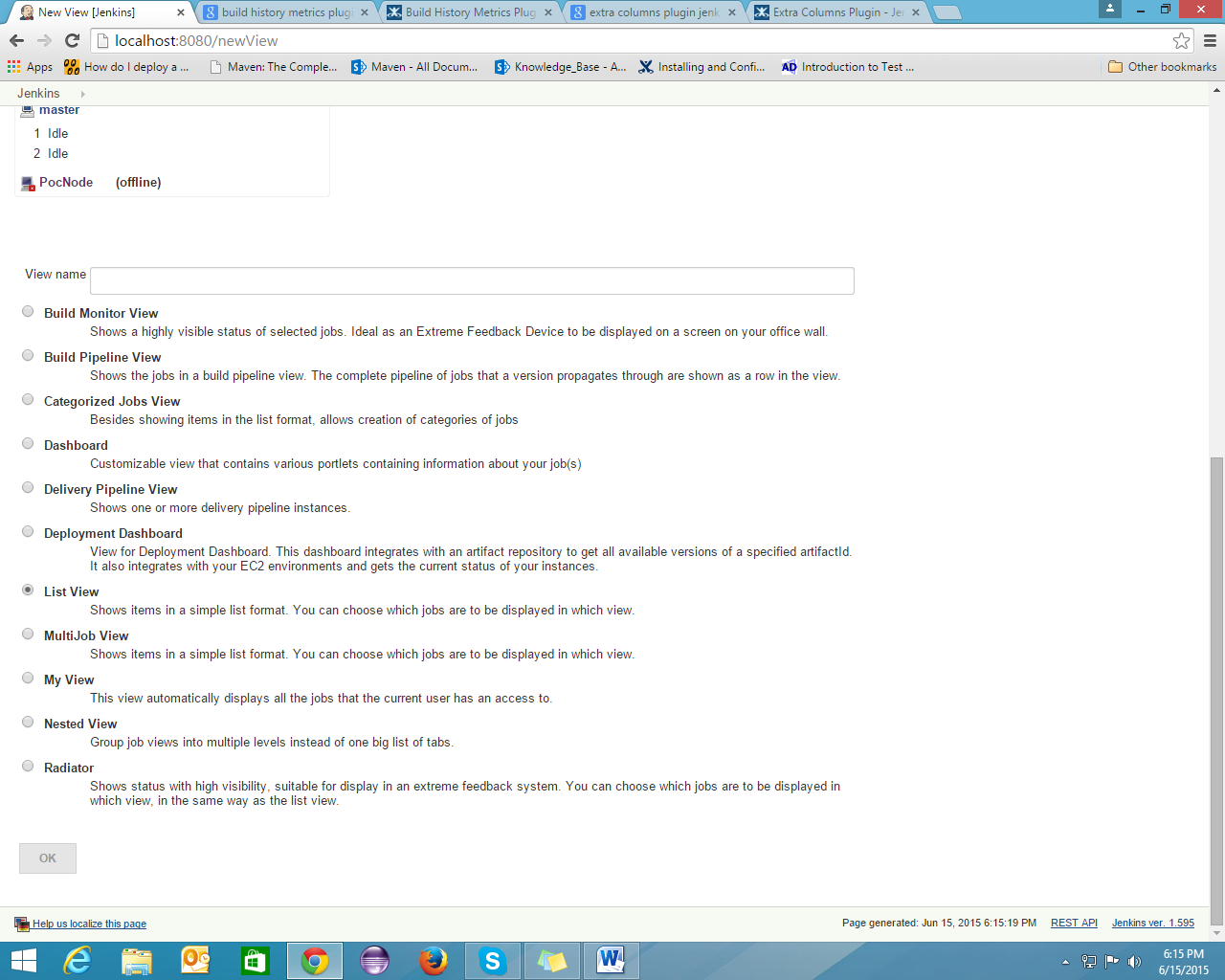
**Extra Column Plugin**

This is a general listview-column plugin that currently contains the following columns: Test Result, Configure Project button, Disable/Enable Project button, Project Description, Build Description, SCM Type, Last/Current Build Console output, Job Type, Build Duration, Build Parameters.

**Configuration:**

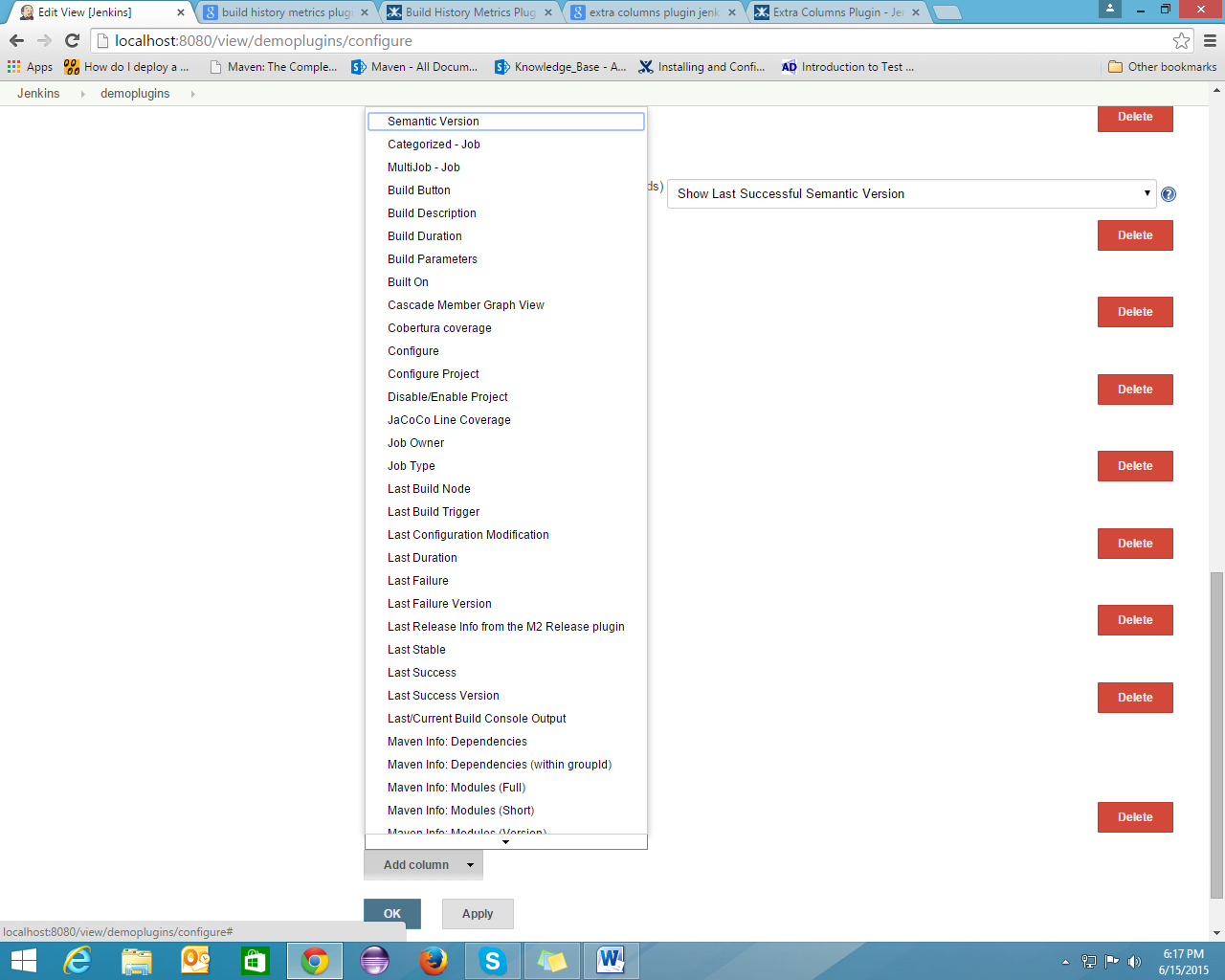
* Create a new view by clicking on the "+" tab, select list view as type and name it. At the bottom of the configuration page of the new view you can add more columns and change their order.





Columns can be configured on the "Edit view" page.

If you want to add columns, you can add columns by clicking on ADD COLUMNS tab.

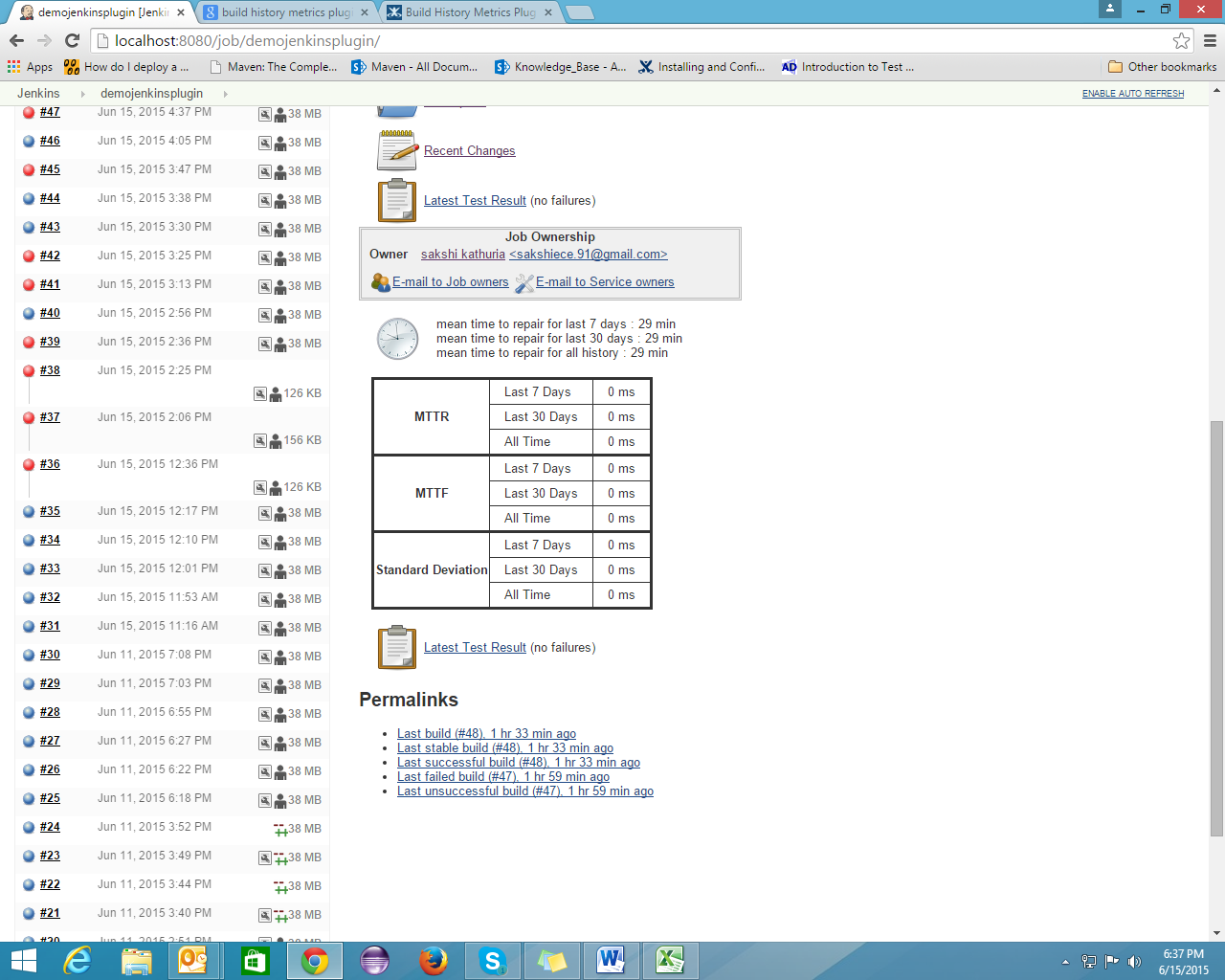


**Build History Metrics Plugin**

This plugin calculates the following metrics for all of your builds once installed:

* Mean Time To Failure (MTTF)
* Mean Time To Recovery (MTTR)
* Standard Deviation of Build Times

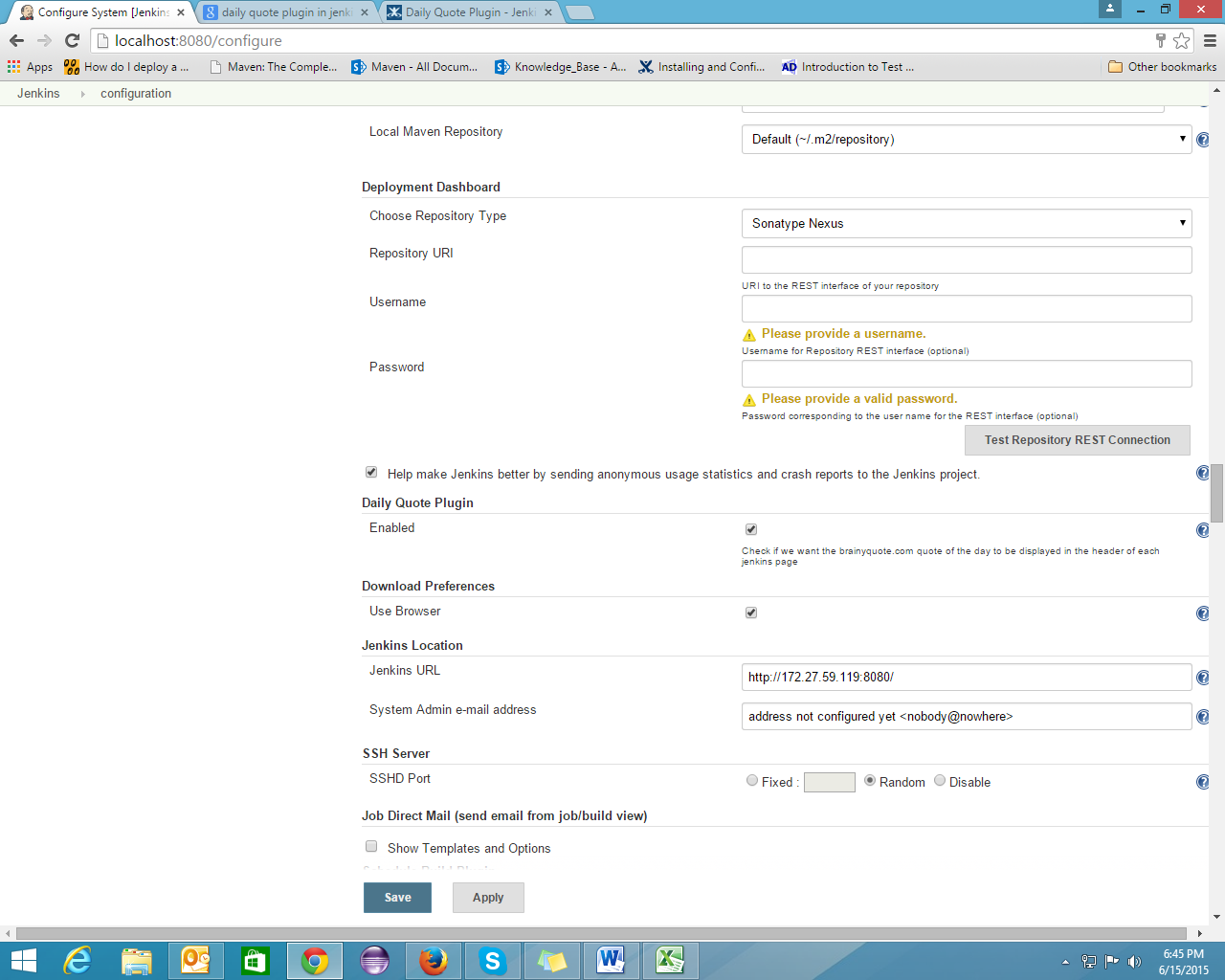
The calculated metrics are displayed in a table on each jobs page showing the metric for the last 7 days, last 30 days and all time.

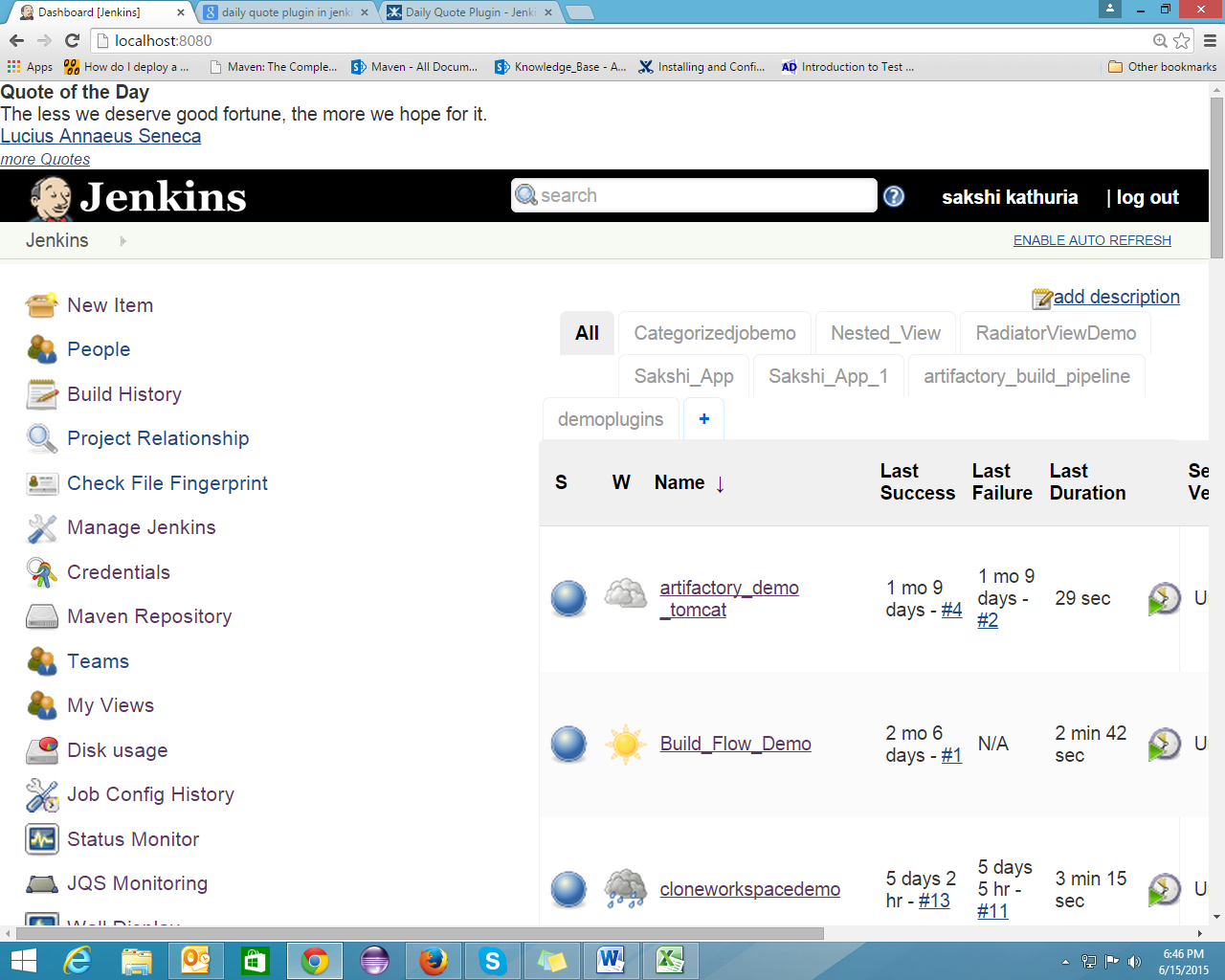


**Daily Quote Plugin**

This plugin will display the brainy quote.com "Quote of the day" in each page header.

**Configuration:**

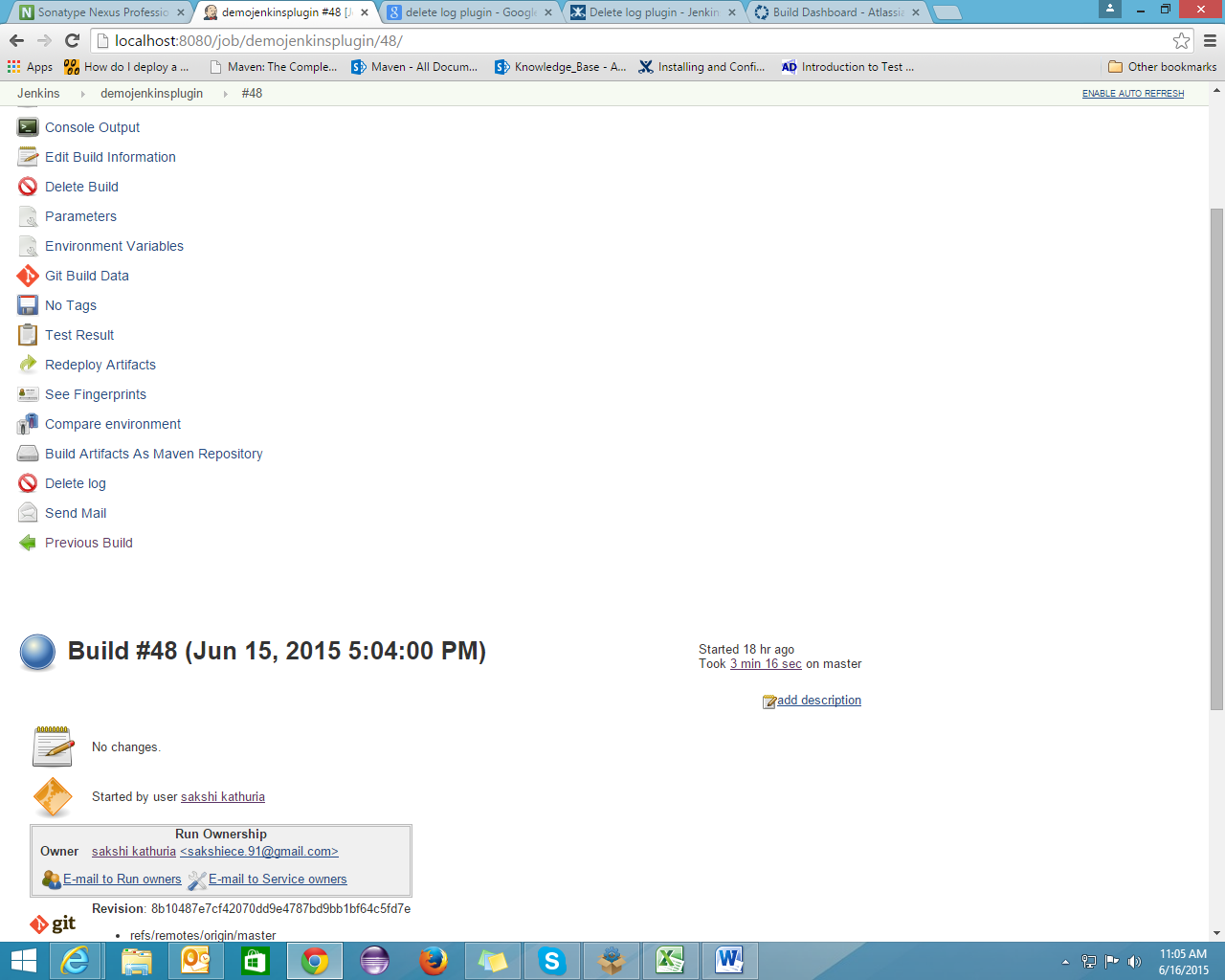




**Delete Log Plugin**

Delete log of a build. In case of matrix build, delete all logs of its configurations too.

**Configuration:**

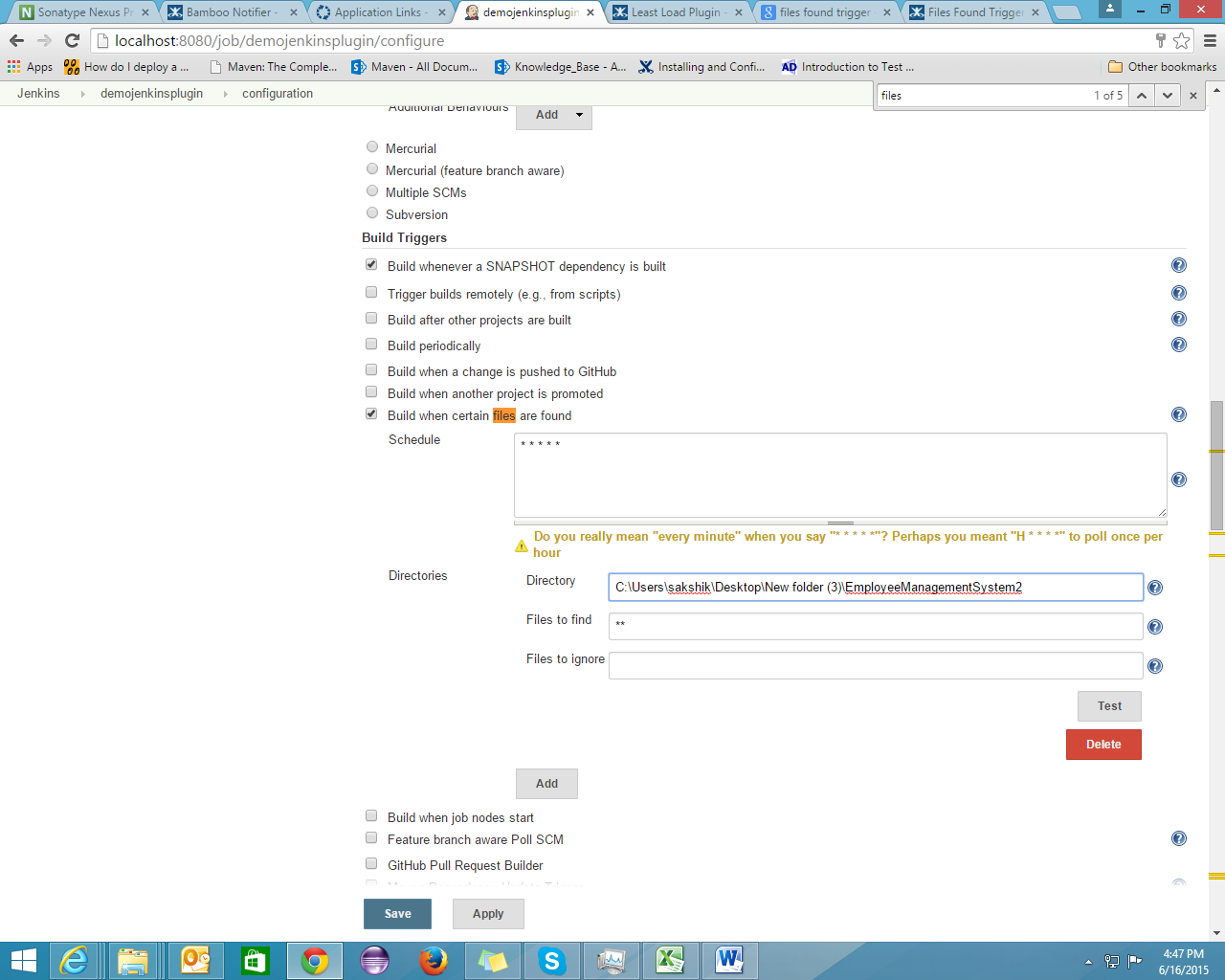


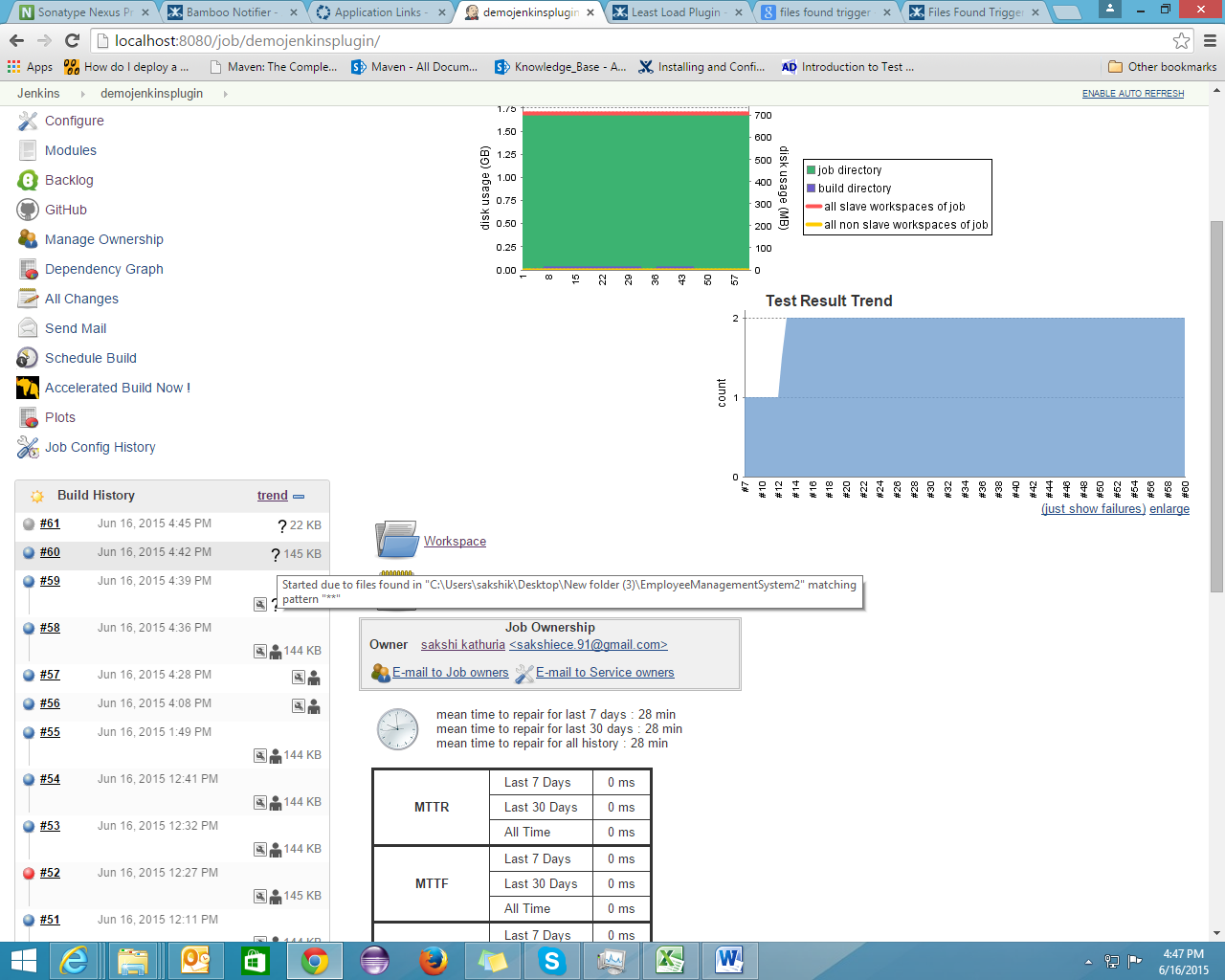
**Files Found Trigger**

Build trigger that polls one or more directories and starts a build if certain files are found within those directories.

Comaptible with java6 and above.

**Configuration:**





**Least Load Plugin**

This plugin overrides the default Load Balancer behavior and assigns jobs to nodes with the least load.

By default Jenkins tries to allocate a jobs to the last node is was executed on. This can result in nodes being left idle while other nodes are overloaded. This plugin overrides the default behavior and assigns jobs to nodes with the least load. The least load is defined as a node that is idle or the one with the most available executors.

**Configuration:**

