JIRA Based deployment Automation

Contents:

[Objective: 1](#_Toc59197770)

[Purpose:](#_Toc59197771) 1

[Overview:](#_Toc59197772) 1

[JIRA & Jenkins Integration Steps:](#_Toc59197773) 2

[Jenkins & Github Integration Steps:](#_Toc59197773) 5

[Jenkins & Artifactory Integration Steps:](#_Toc59197773) 6

[References:](#_Toc59197773) 8

Objective:This document is desired to showcase the JIRA – based deployment automation through CI / CD devops tools integration (Jenkins, JIRA, GitHub, Artifactory)

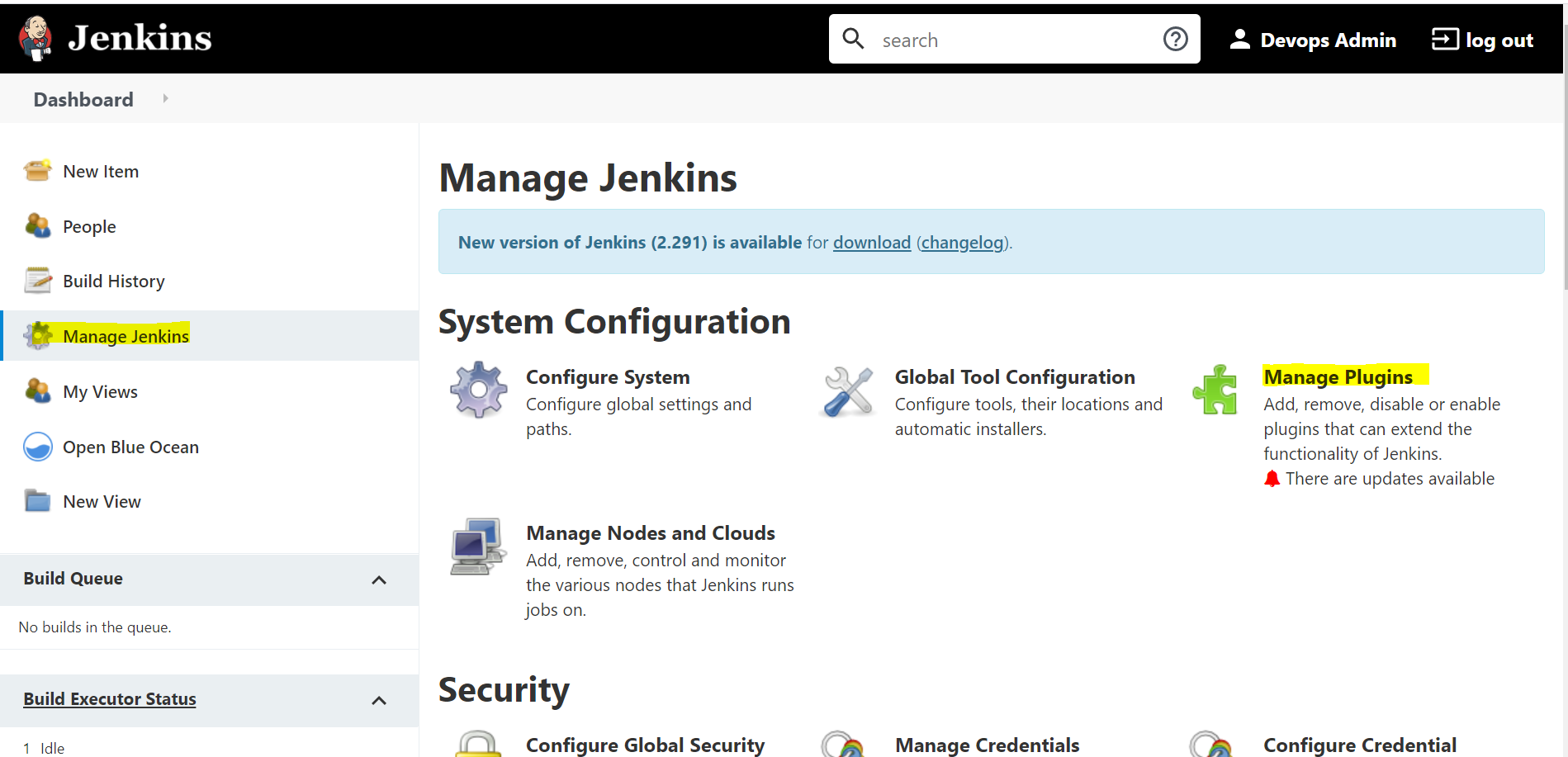
# Purpose: Now entire world is on agile implementation with devops -automation, This case study is one of such reference to kick off deployments with a one click flow straight from the issue ticket (JIRA), Any new project which is planning for a devops CI / CD setup can consider this case study and reuse & implement this In the deployment flow.

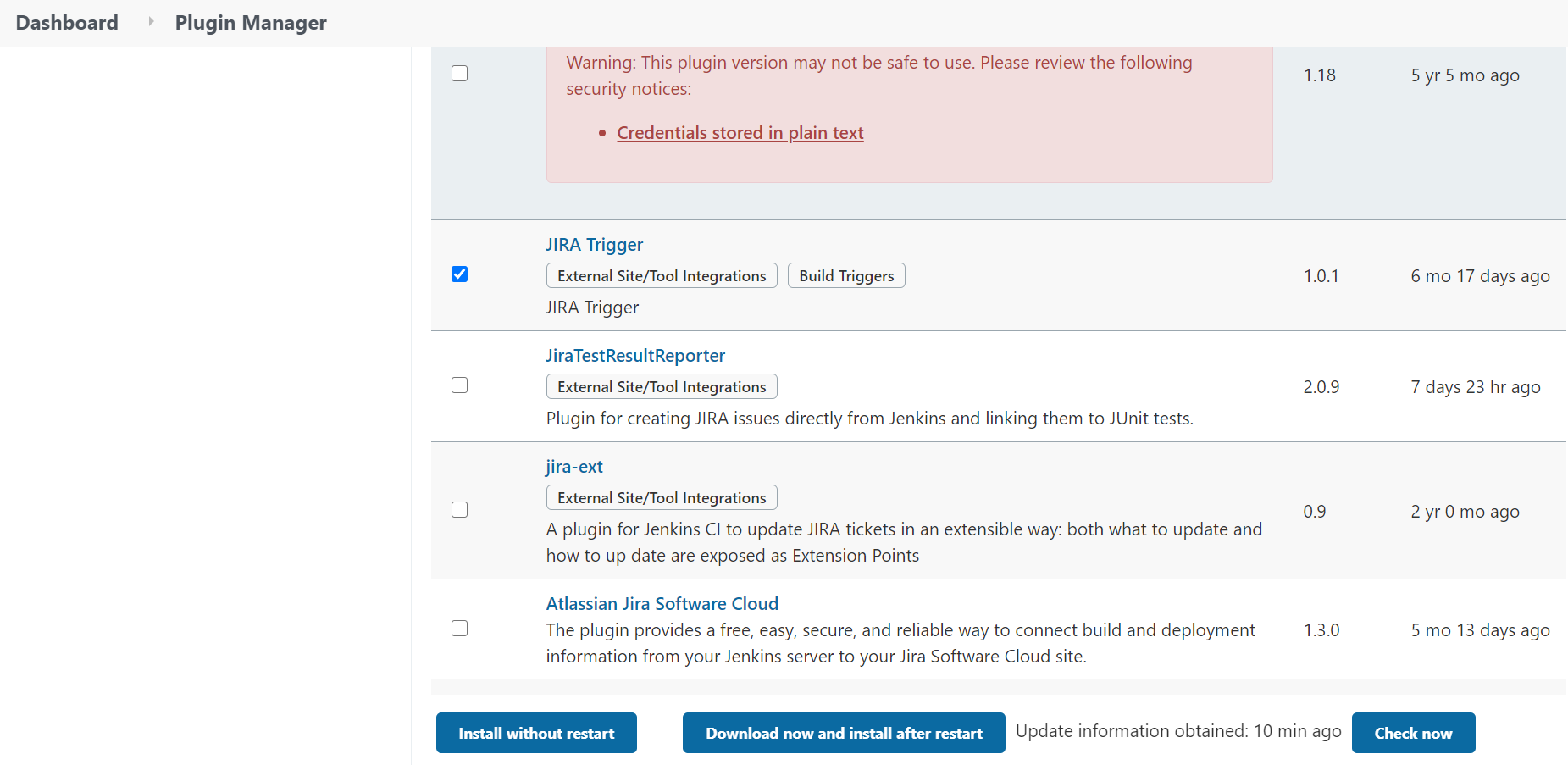
Case Study Overview:

1. JIRA: Jira Software is an agile project management tool that supports any agile methodology, be it scrum, Kanban, or your own unique flavour.
2. Jenkins: An open source automation server which enables developers around the world to reliably build, test, and deploy their software.
3. Jenkins- JIRA – Integration: This integration exists to provide a free, easy, secure, and reliable way to connect your Jenkins server, running behind the firewall, and Jira Software Cloud and/or Jira Service Management Cloud. Simply add a few extra lines in the *Jenkins file* that defines your pipeline steps and the integration will start sending build and deployment data to Jira and associate that information with the relevant Jira issues.
4. JENKINS – Build Trigger from JIRA: JIRA & Jenkins integration provides a excellent feature of controlling the Jenkins builds (deployments) right from the actions of the JIRA issues / tickets. This involves Jenkins plugins (JIRA & Jira Trigger), the base of build control actions from JIRA is based on JQL (JIRA query language).
5. Some of the Key features:
6. Triggers a build when a comment is added to JIRA
7. Triggers a build when an issue is updated in JIRA
8. Reply to JIRA for scheduled builds
9. Sets JIRA information as environment variables to the triggered build.

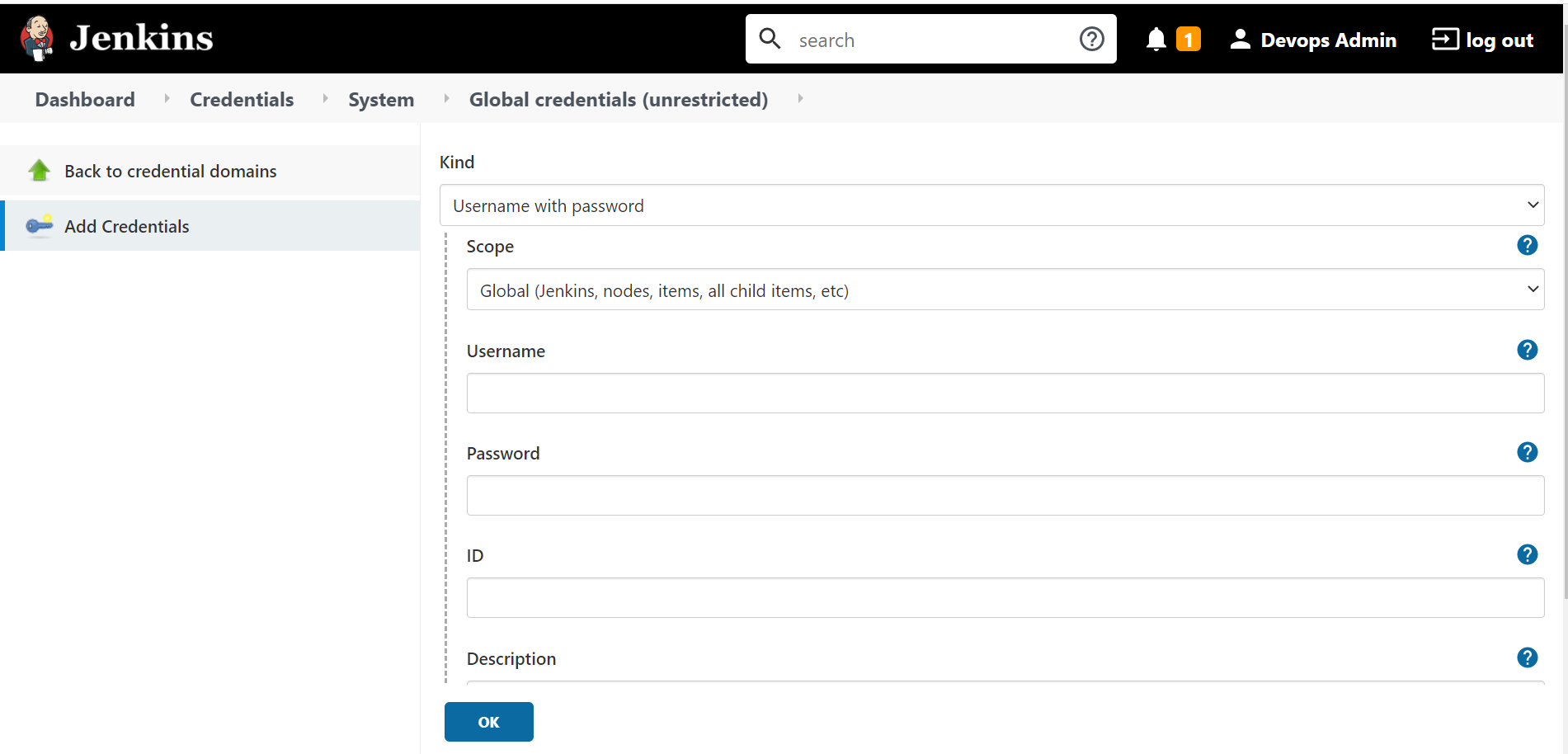
Jenkins & JIRA Integration Steps:

1. Install JIRA plugins in Jenkins.
2. Steps: Jenkins home Page-> Manage Jenkins -> Manage Plugins ->Search for JIRA -> Select JIRA & JIRA trigger plugins ->Install Without restart.

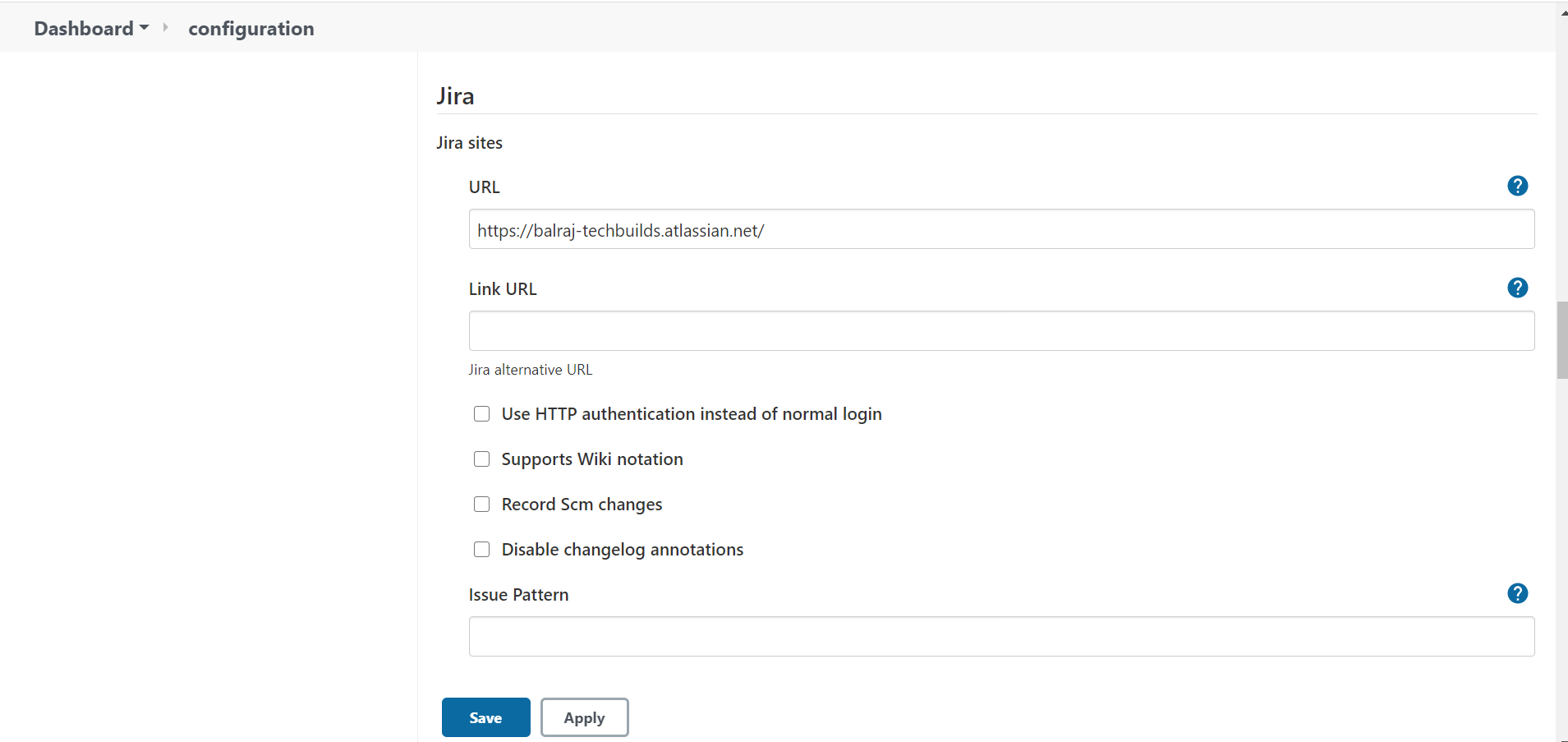


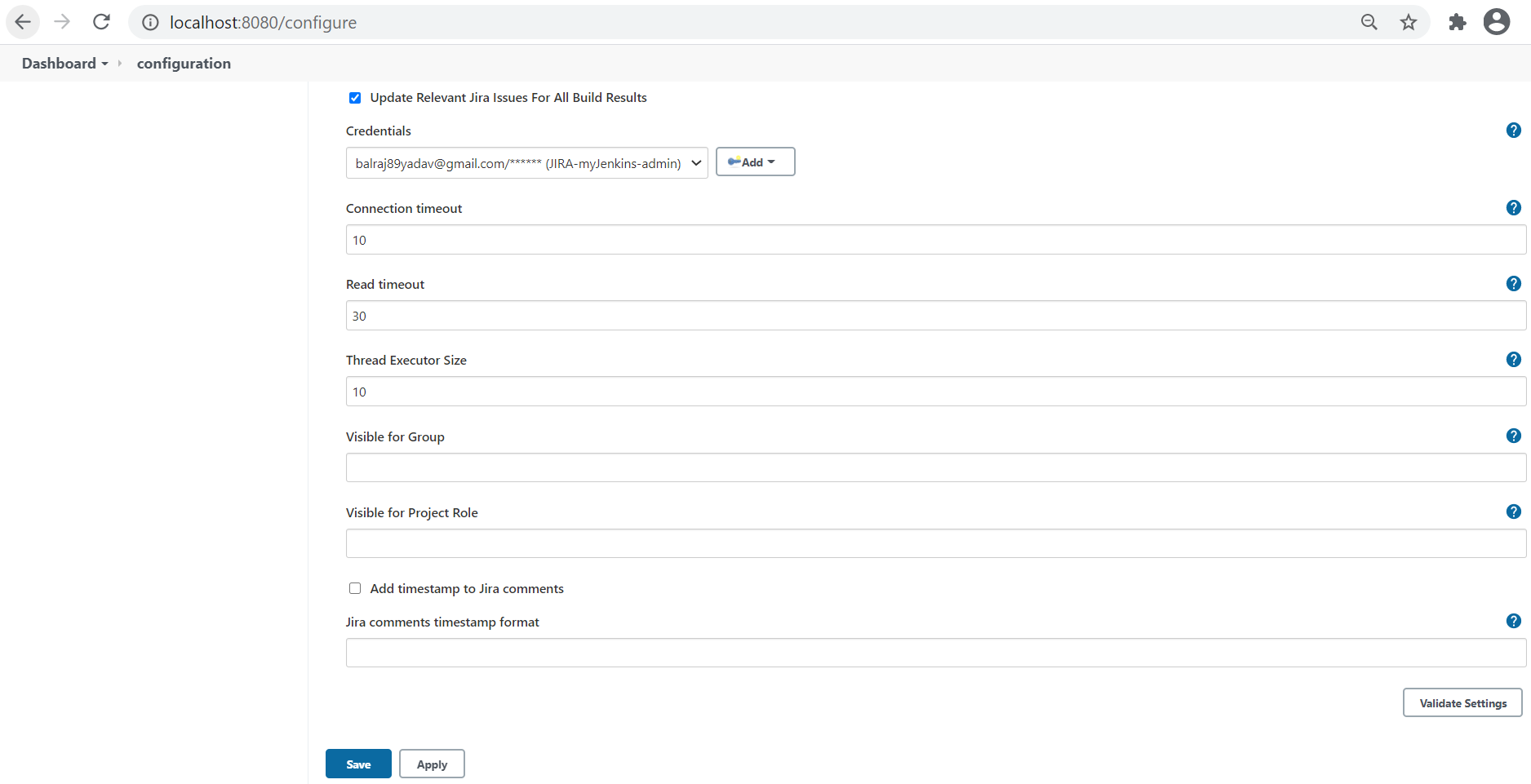


1. Configure JIRA credentials in JENKINS:
   1. Steps: Jenkins Home Page -> Manager Jenkins-> Manager Credentials -> global Credentials -> add credentials->Provide JIRA login details info as share in below screen.
   2. Username: JIRA username
   3. Password: Should be API key which need to be copied from the JIRA portal
   4. ID: any unique identification of this global credentials.
   5. Description: Specific to the credentials

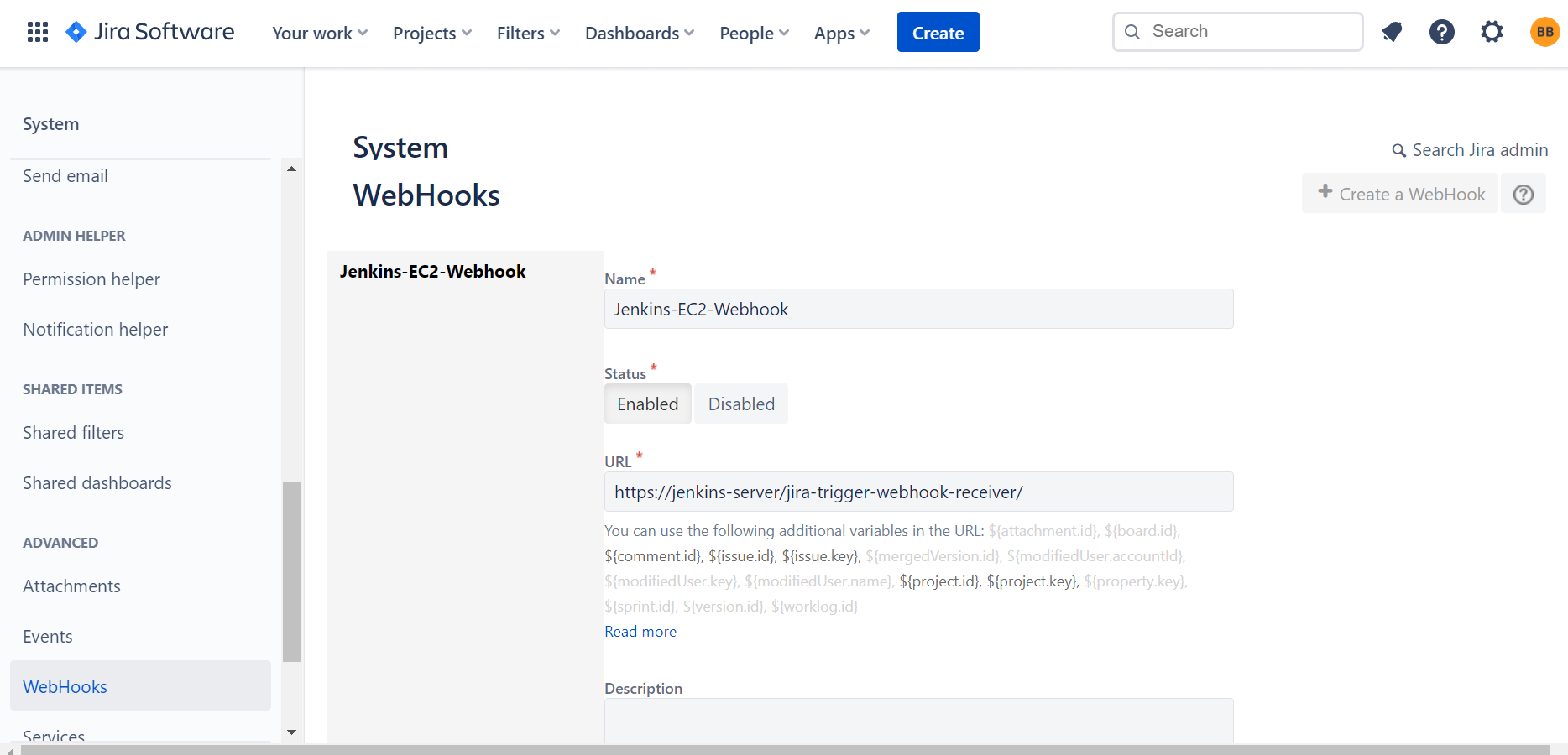


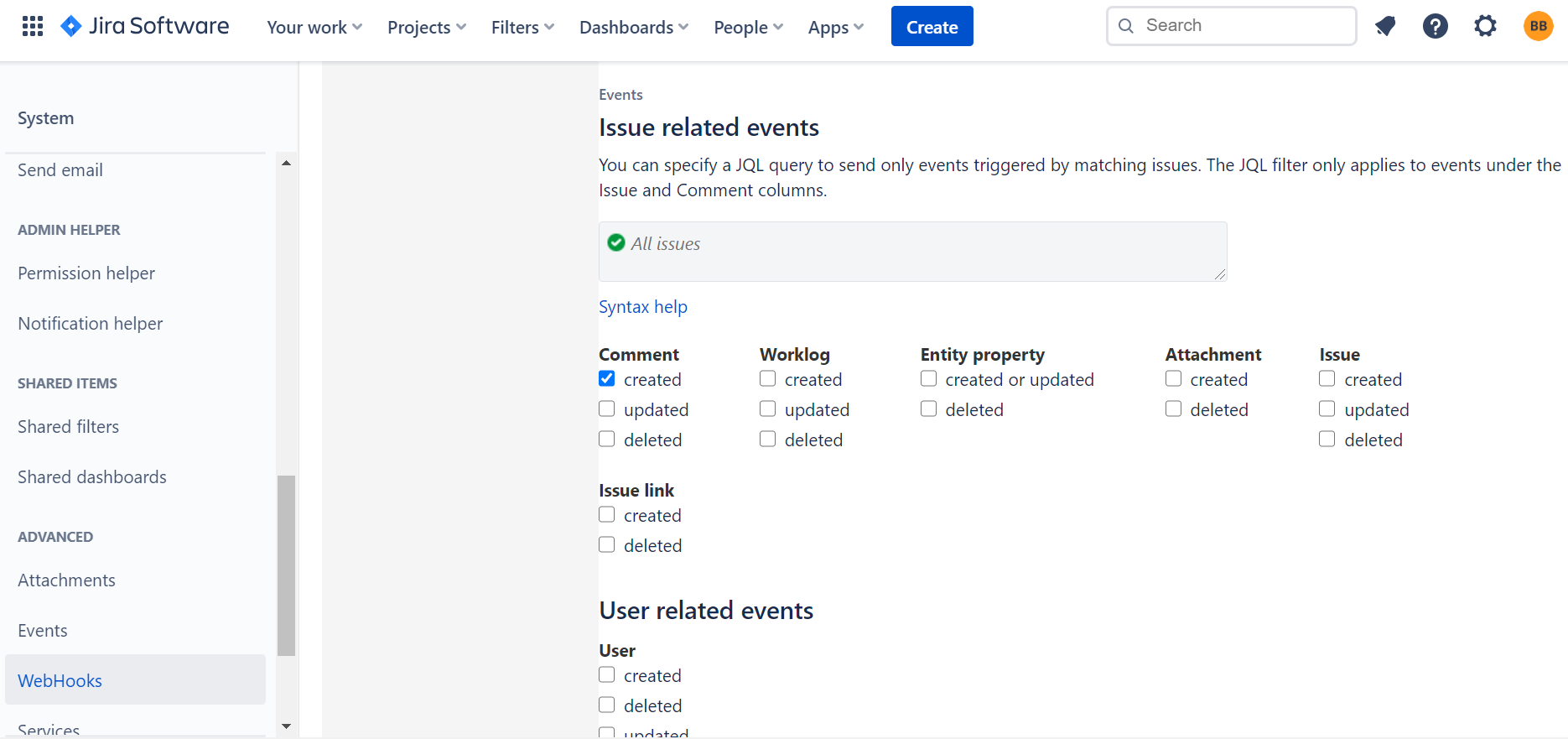
1. Setup JIRA URL in Jenkins:
   1. Steps: Jenkins home -> Manage Jenkins -> Configure System -> go to JIRA sites section-> add->Configure
   2. URL : URL should be your JIRA running URL ex : <https://Example.atlassian.net>
   3. Check the “Update Relevant Jira Issues for All Build Results”
   4. Select JIRA credentials from the dropdown, you should see this in dropdown since you have already configured JIRA credentials in global configuration credentials as done in above section, if Jira credentials not exist, you can reconfigure using add button again.
   5. After this Validate the settings, a success return shows your JIRA integration success
   6. Apply & Save.

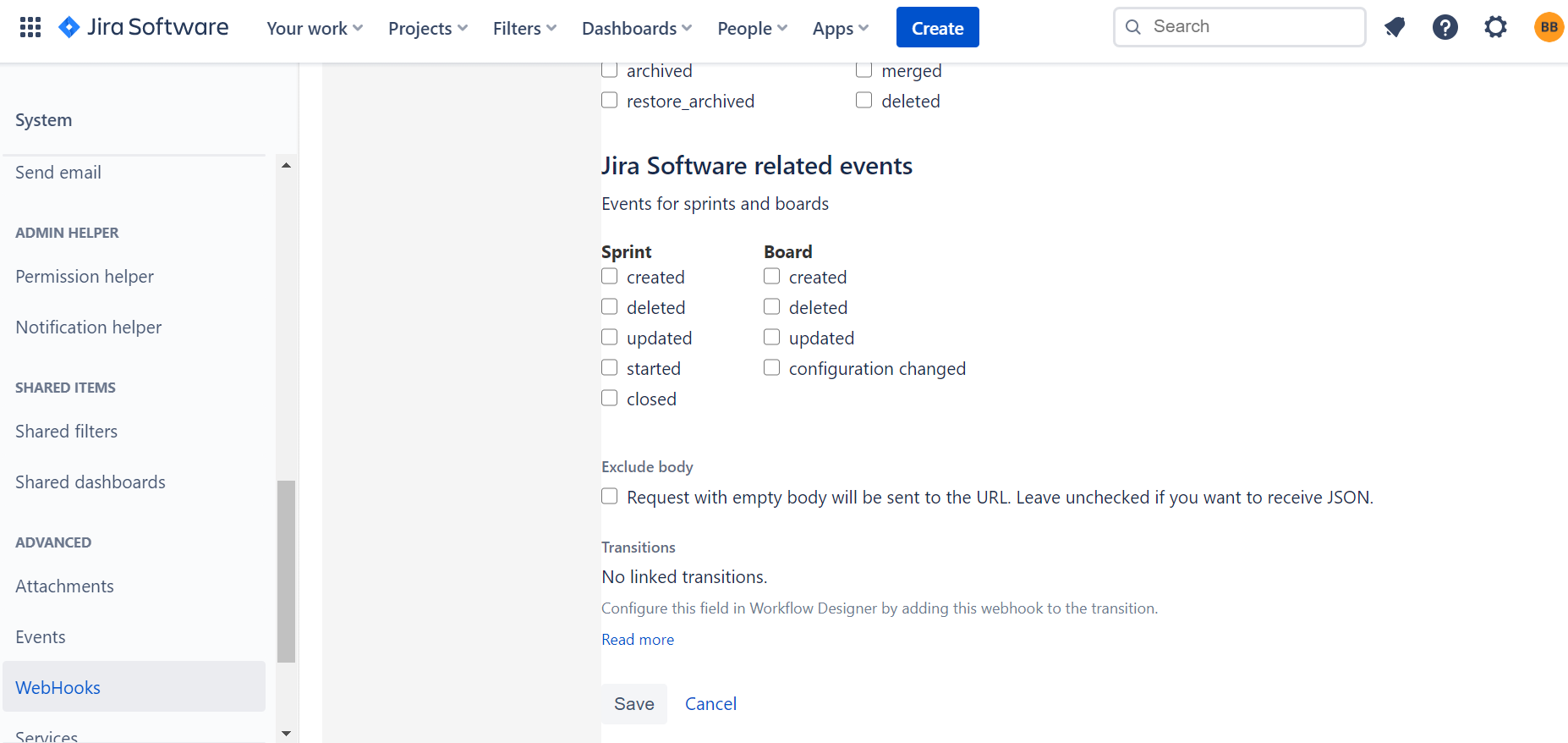




1. POST Webhook configuration in JIRA.
   1. Note: You should be having administrative privileges to perform webhook configuration in Jira portal.
   2. Steps: Settings -> System -> Select Webhooks -> Create New webhook
   3. Name: Please provide any specific name to this webhook configuration
   4. Status: should be enabled if this webhook to be in active state.
   5. URL : Please provide your Jenkins server URL appending with “jira-trigger-webhook-receiver”, Ex : <https://jenkins-server/jira-trigger-webhook-receiver/>
   6. Issues Related: Can select all issues
   7. In the next section you can select any event based on requirement, example comment created issue updated etc.
   8. Note: Do not check the exclude body check box.
   9. Done Save the webhook.
   10. Note: Jenkins URL should be https and running on 80 & 443 if using JIRA cloud server and no issues with port or https if using JIRA server.

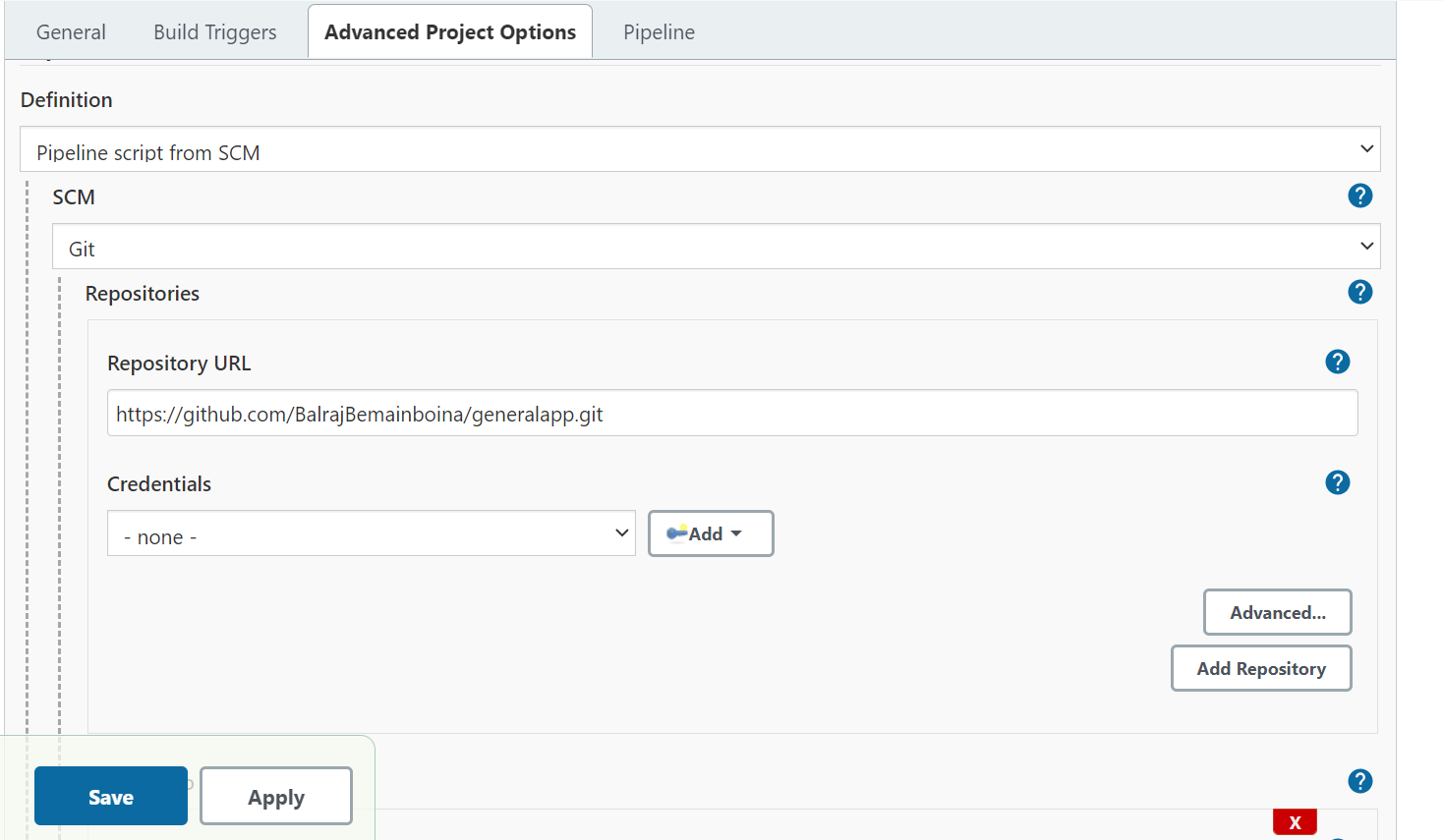






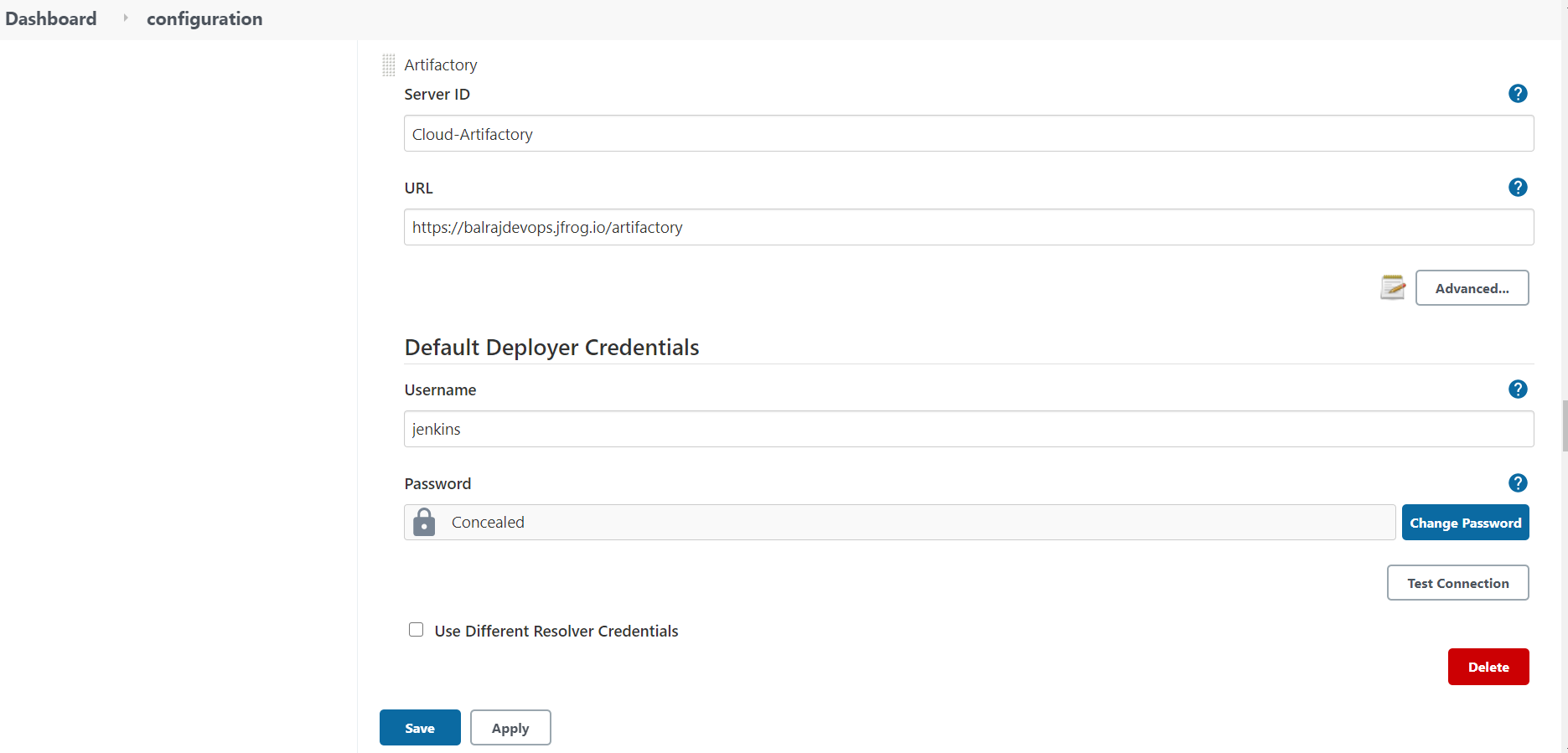
Jenkins & GitHub Integration Steps:

1. Need to have GitHub, git plugins to be installed in Jenkins server.
2. Dashboard -> Manage Jenkins -> Manage plugins -> search for git -> install
3. Configuring the GitHub repo can be done right in the job configuration example below for a pipeline job.
4. configure -> Pipeline from SCM:
5. SCM: git
6. Repository URL : Mention the repository URL(ex: <https://github.com/example/example.git>)
7. Credentials: select credentials if configured globally or add credentials.



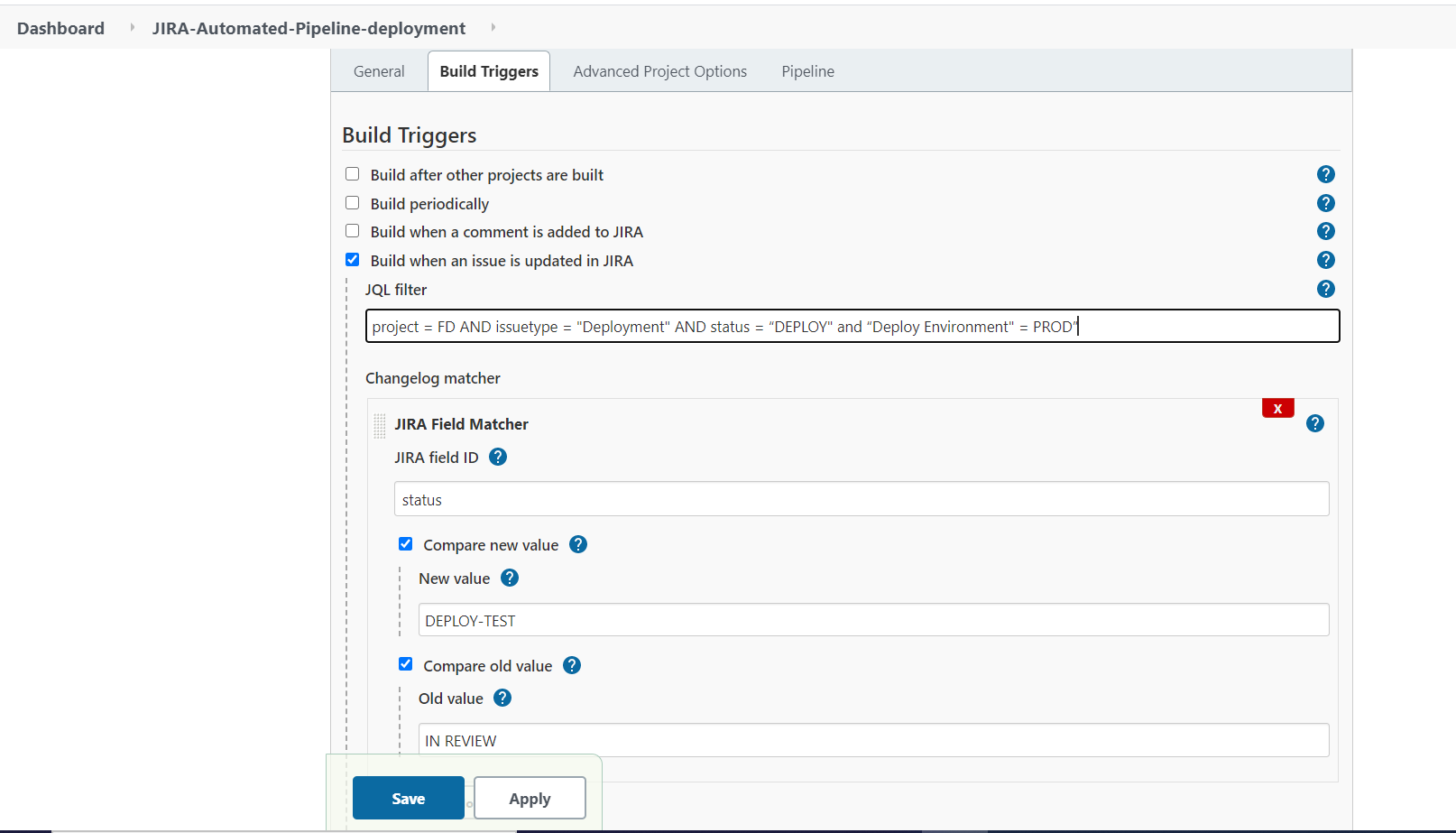
Jenkins & Artifactory Integration Steps:

1. Need to have Artifactory cloud account or Artifactory server and artifactory plugin to be installed in Jenkins.
2. Dashboard -> Manage Jenkins -> Manage plugins -> search for artifactory -> install.
3. For configuring artifactory post plugin installation in Jenkins got to Manage Jenkins-> configure system -> Go to Artifactory server section.
4. Server ID: is the Unique identification (Name) to be given for artifactory server so this can be used across all the jobs globally.
5. URL: Artifactory server URL.
6. Default Deployer Credentials: Login credentials of the artifactory server.
7. Username & password: artifactory username & password.
8. Do a test connection, If the configuration is correct it should return the artifactory version of server connected.
9. Note: It is suggested to create a Jenkins specific admin account in artifactory.

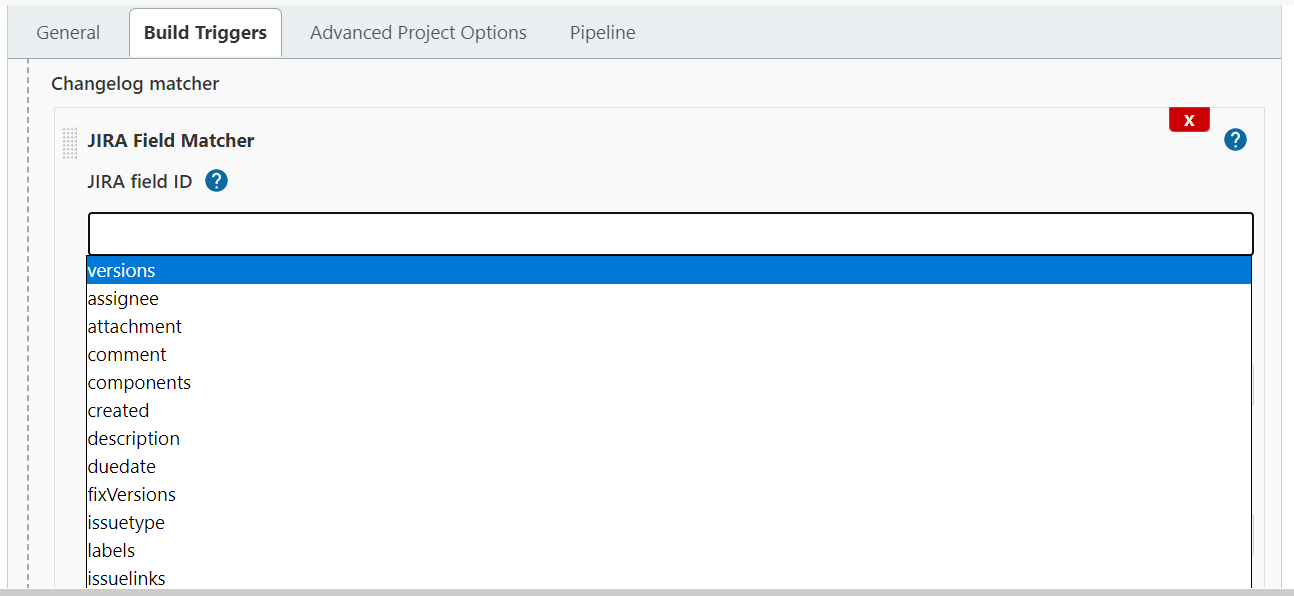


JQL- JIRA Query Language:

1. JQL – JIRA query language is a statement based [search for your issues in Jira](https://twitter.com/intent/tweet?source=webclient&amp;via=atlassian&amp;text=search%20for%20your%20issues%20in%20Jira&amp;url=https://www.atlassian.com/blog/jira-software/jql-the-most-flexible-way-to-search-jira-14) database. JIRA server will execute the query language statement which is mentioned and based on the condition frame it queries the database of jira and responds back with the issue details.
2. As any other language JQL also uses the general ‘AND’ ‘OR’ ‘IS’ keywords and operators for framing the query statement.
3. Ex:” project = FD AND issuetype = "Deployment" AND status = “DEPLOY" and “Deploy Environment" = PROD”
4. Above given example is a JQL filter which points to project FD and if the issuetype is deployment and when the issue status is deploy and environment is prod, when the criteria is met it returns the ticket details.
5. In Jenkins once the JIRA trigger plugin is installed when you are trying to new job the build triggers section will automatically populate the trigger option of based on JIRA updates, which also has JQL filter which provides a option to build the trigger once the JQL requirement is met.
6. Also, you can mention the triggers by configuring the field matcher section.



1. Above screenshot shows the build trigger section options to trigger the builds based on the JIRA comments update of JIRA issue update, In the issue update you can mention the JQL filter or JIRA field matcher selection.
2. Further JIRA field matcher has multiple options to select based on the requirement.
3. For More on JQL please visit: https://www.atlassian.com/blog/jira-software/jql-the-most-flexible-way-to-search-jira-14



References:

1. Direct Download of Jenkins Plugins : <https://updates.jenkins-ci.org/download/plugins/>
2. JIRA query language: <https://www.atlassian.com/blog/jira-software/jql-the-most-flexible-way-to-search-jira-14>
3. JIRA-Trigger plugin info: <https://github.com/jenkinsci/jira-trigger-plugin>
4. JIRA plugin info: <https://github.com/jenkinsci/jira-plugin>
5. Jenkins Artifactory plugin: <https://github.com/jfrog/jenkins-artifactory-plugin>
6. Jenkins git plugin : <https://github.com/jenkinsci/git-plugin>
7. Jenkins Pipeline Examples : <https://github.com/jenkinsci/pipeline-examples>