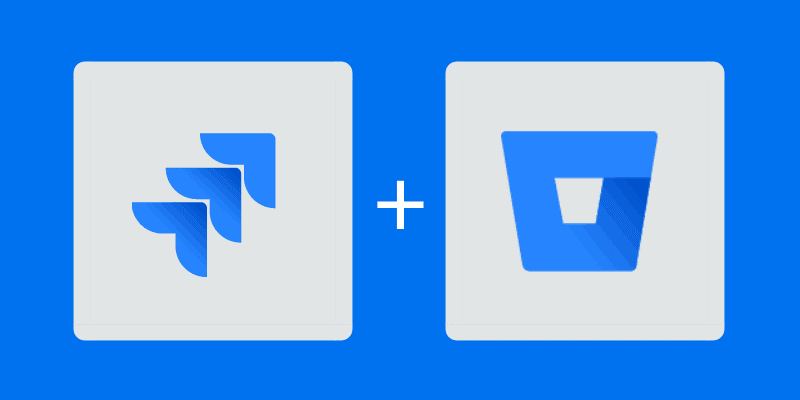
**Complete guide to Jira and Bitbucket software integration**

* [**INSTALLATION**](https://www.ratosan.com/category/installation-services/)
* [**MEHDI HEYDARI**](https://www.ratosan.com/author/mheidari/)

**Table of Contents**[ [Show](https://www.ratosan.com/jira-and-bitbucket-integration/) ]

4.8 / 5 - (11 points)



A software product goes through various stages such as planning, development, testing and publishing to reach its customers. In the meantime, various processes, tools, standards and methodologies are used to create a better product.

Jira is one of the most widely used software project management tools that has provided excellent capabilities in this field. Many companies around the world use JIRA software to manage and track their work and projects. These companies use JIRA software to schedule, track, record work schedules, and schedule upcoming tasks. This has enabled many tools to integrate with Jira software.

Software developers use a variety of tools to maintain and manage their code. One of the best tools in this field is Bitbucket. Bitbucket software, which is a very useful Git-based Source Controlling software, is widely used in companies around the world.

The close connection between the planning and development stages of a product requires the need for a proper and integrated platform for these matters. Jira and BitBack software, which are the best software in these two fields, have provided full integration capability, and by integrating Jira and BitBack software, it will be possible to combine management processes and software development processes.

BitBack and Jira software are both Atlassian products, which simplifies the process of integrating the two software. Both software are fully mature and focused on helping software developers.

In this article, we are going to describe how to integrate these two widely used software and explain how these two softwares help modern software development teams.

**Types of bit bucket versions**

Before starting the integration of these two softwares, it is necessary to mention one point. Two versions of Bitbucket software are provided by Atlassian. Server version (Bitbucket Server) and cloud version (Bitbucket Cloud). There are differences between the two versions.

In the cloud version of BitBack software, everything is done by Atlassian and you only need to work on the code and repositories and manage them. The BitBact server version will be more suitable for teams that need more control. Because in the server version, teams will have all the settings of this service. In this version you will install and manage the software on your servers. If you need help installing software, go to [our contact page](https://www.ratosan.com/contact-us/) and get in touch with our experts.

According to these explanations, in this article, we have tried to make all the contents usable in both server and cloud versions. But most of our focus in this article is on the server versions of both BitBack and Jira. [Contact us](https://www.ratosan.com/contact-us/) to install and set up Jira and Bitbakt systems as servers .

**Why should we integrate Jira and Beitbakt together?**

The first question we need to ask ourselves here is what is the need to integrate these two systems together and what will be the benefit of doing so?

There are many reasons for integrating these two systems. In the first place, the integration of these two systems will provide us with useful features after integration. As we will see, by integrating these two systems, very good new features will be activated.

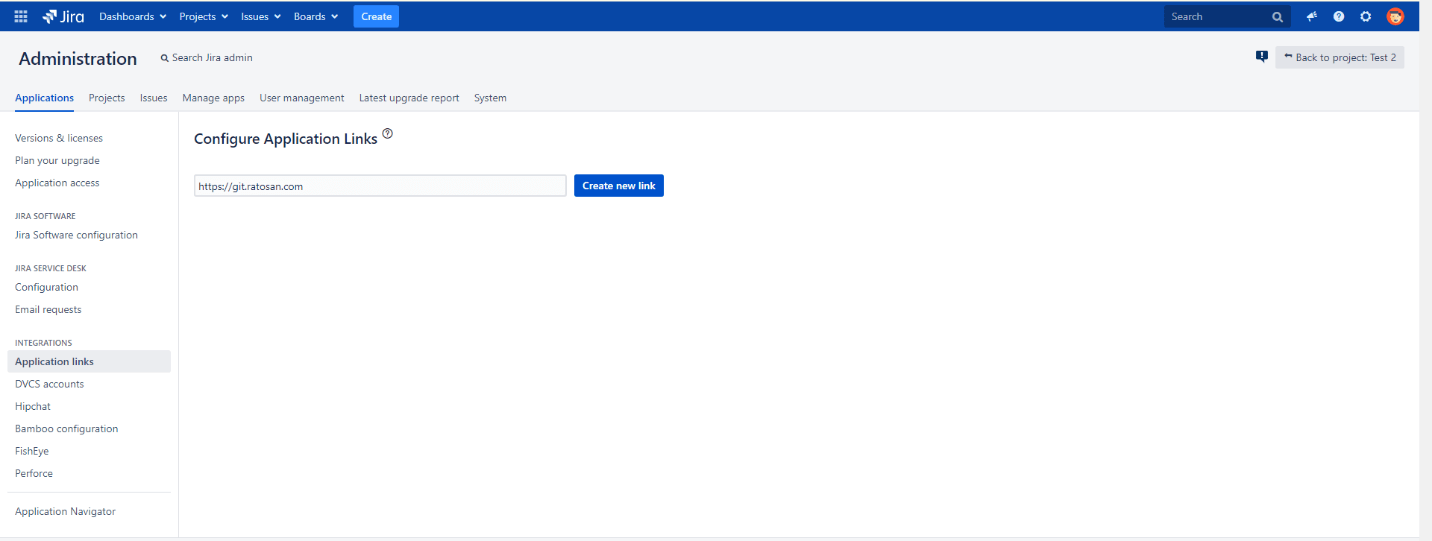
Many teams use JIRA software to manage their software development processes. By integrating Jira and BitBack software, a complete view of the software environment and software development process will be provided to the team members.

**Start integrating Jira and Bitbakt**

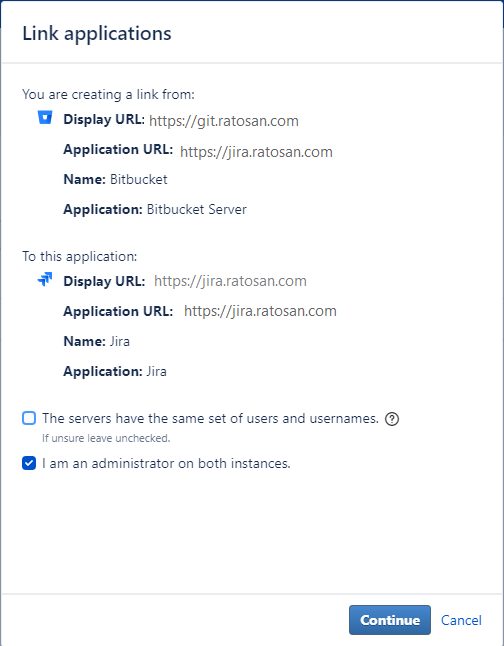
You can easily do this once you have decided to link BitBack and Jira. To do this, enter one of the BitBack or Jira software. Select Application Link from the Administration section.

For example, in Jira software, go to the Application section from the Administration section and enter the Application Link.

After entering the URL of the BitBact system, click on Create new link.

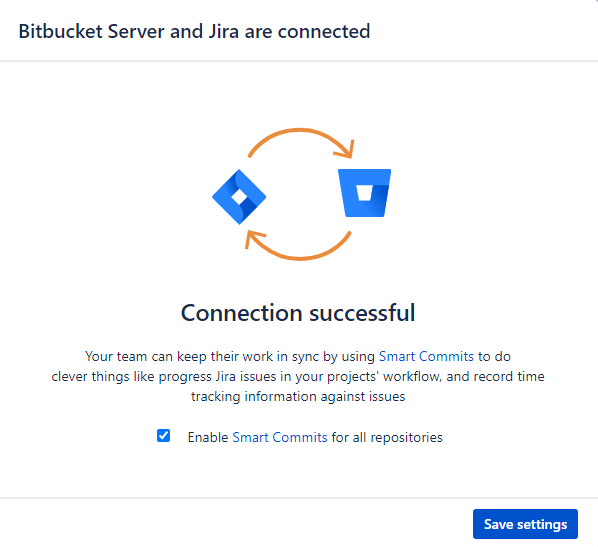


If the JIRA system has the necessary access to connect to the BitBact service, the following window will be displayed and the JIRA system will receive and display the BitBact system information.



If the username and password of the Administrator user are the same in both systems, select the option The servers have the same set of users and usernames. Enter.

If the integration is successful, the following screen will be displayed to save the Smart Commits settings.



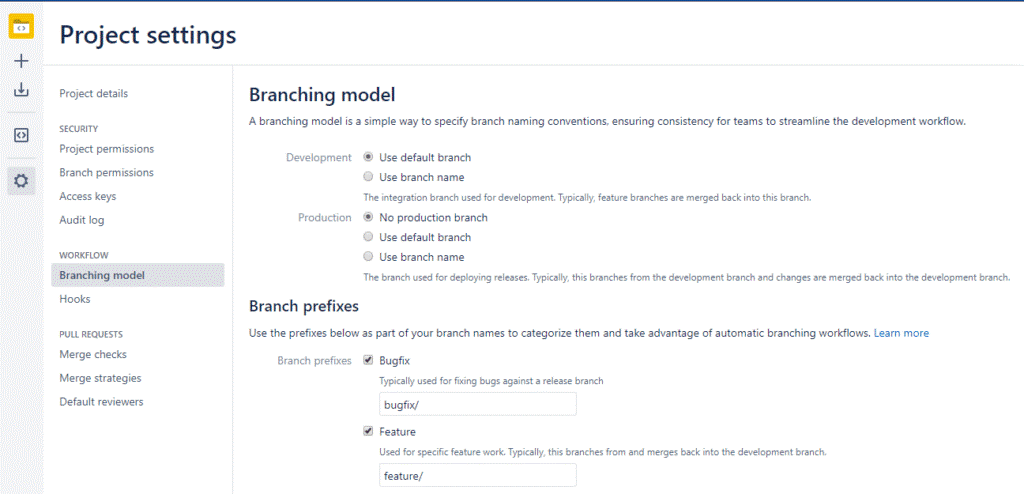
After completing the integration process of the two softwares, the development section will be displayed in Jira's icons.

**Optimizing BitBact to work with Jira**

So far, you've done the initial integration of the two software. For the two software to work better together, you need to make small changes to the BitBact software to work with your needs.

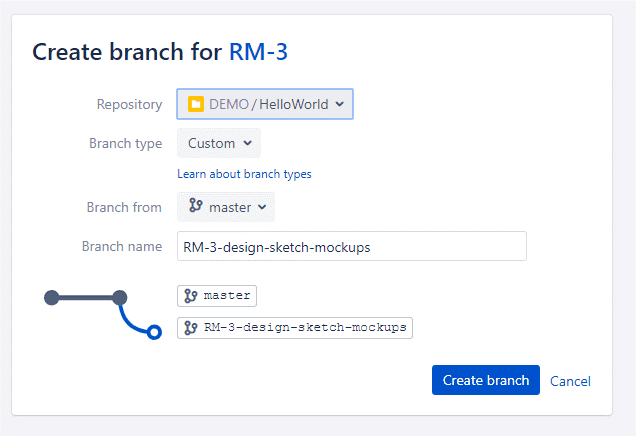
To get started, go to the Branching model section in the BitBacket software to make the appropriate settings. In this section, you will set up how to perform Branches in BitBack software. This feature allows you to specify how to name branches. Also specify what the merging policy is.

For example, in the settings of this section, you can set the types of branches that can be created and the name of each branch. Merge policies are also defined in this section.



Why are these settings important? Because by integrating Jira and BitBact software, you will be able to create and manage branches in Jira software as easily as BitBact. Therefore, if you create a branch directly from an issue in Jira, it will be in accordance with the settings of this page.

When a User Story is assigned to a developer in Jira software to implement a new requirement in the software, the user on the page related to the same issue by clicking the create a branch option allows the branch to develop code related to the User Story Create. Clicking this option will open a window for the user in which the necessary settings to create a branch will be visible. These settings are in accordance with the policies you have set in the Branching mode section.



**Smart Commit feature**

A very useful feature that can not be ignored is the Smart Commit feature, which is activated by integrating Jira and BitBact software. This feature, which was launched from version 4.2Bact and version 4.1 Jira, allows the user to enter commands for Jira software by entering specific commands when committing code in the Comment section.

For example, when a developer wants to commit his code, he can close the issue by entering the key for one (or more issues) and then entering #close in the comment of a commit.  
A better feature is that the developer can directly spend the time spent on developing this part of the code by entering the value #time and then entering the amount of time (in a format similar to 1w 2d 3h 15m meaning a week And two days, three hours and 15 minutes), enter Jira. This can also be done manually with log work in Jira.

Most of these commands can be used in Bitbact software. Things that can be used include:  
• Posting a comment on an issue in Jira  
• Recording the time spent on an issue (Jira Time Tracking feature)  
• Moving the Issue from one situation to another based on the flow Job defined in JIRA  
smart commit commands have a structure similar to the following:

<ignored text> <ISSUE\_KEY> <ignored text> # <COMMAND> <optional COMMAND\_ARGUMENTS>

This structure needs to be inserted in the Comment field when the developer commits the code. In this structure, the ignored text sections of the command will be omitted and will have no effect on Jira, and will be entered only to describe the description of this Commit. The following are examples of using the smart commit feature.

**Enter a comment in an Ishu:**

Some text JRA-34 #comment corrected indent issue

In this command, a comment with the value of corrected indent issue will be entered in the JRA-34 icon in JIRA software.

**WorkInsert Work log in an Ishu:**

JRA-34 #time 4h 30m Total work logged

In this command, a work log with a time of 4 hours and 30 minutes is entered with the explanation of Total work logged in the logger for IRA JRA-34.

**Ishu transfer in workflow:**

JRA-090 #close Fixed this today

In this command, the IRA JRA-090 has been transferred to another state with a transition called close and with the explanation Fixed this today.

Using the smart commit feature is not limited to these three modes, and you can extend the use of this feature by installing related plugins. One of the useful plugins in this field is [Smart Commits for Bitbucket](https://marketplace.atlassian.com/apps/1212969/smart-commits-for-bitbucket?hosting=server&tab=overview) .

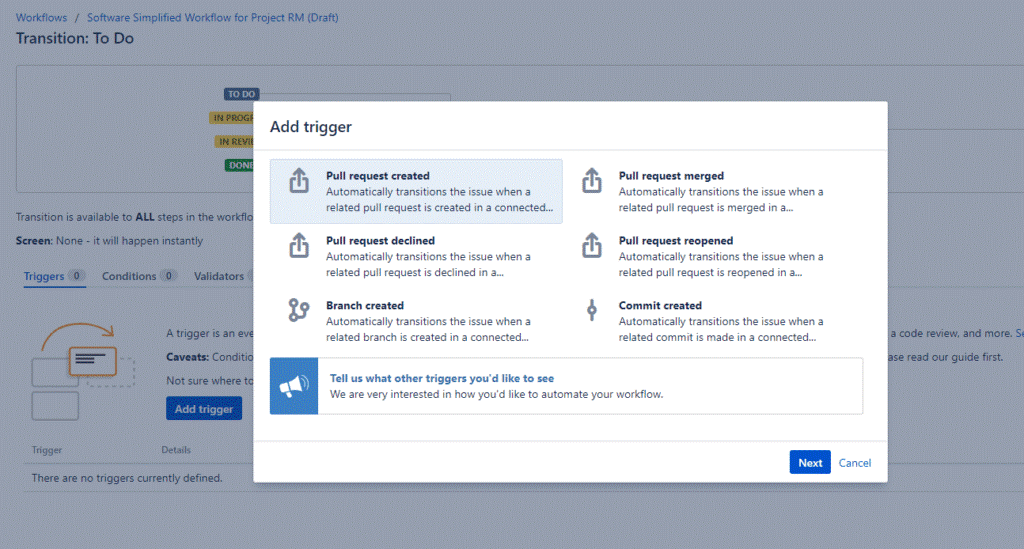
**Add bitobact to Jira workflows**

Once you have made all the integration settings in BitBact, the settings can be implemented and used in Jira software.

Another feature we are going to look at here is the ability to change the status of an Isho in Jira based on events that occur in BitBack repositories.

For example, if a pull request is made to a branch in Bitbact, the corresponding ichos will be moved from In Development to Code Review.

This will be done by making settings in the Trigger section of a workflow in Jira.



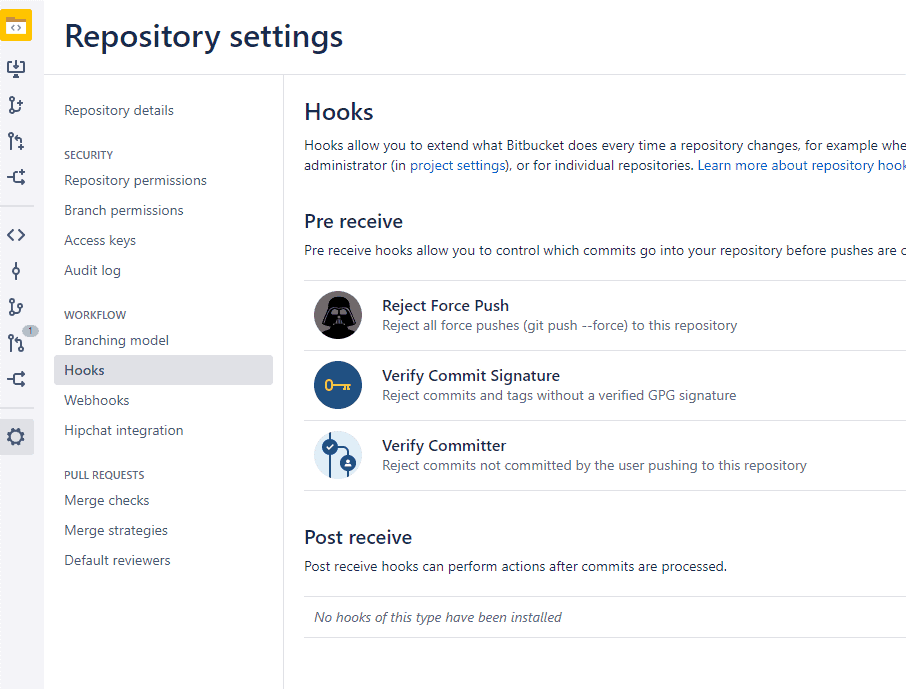
Trigger feature in Jira workflows is a powerful tool that allows you to automatically synchronize the status of objects in a JIRA project with the information of your software development tools (such as BitBact).

In fact, instead of the software developer manually updating the status of the associated objects after tasks such as commit code, code review, create a branch, and صورت manually, you can configure the triggers of a workflow to Automatically activate activities.

After making the Trigger settings in the Jira workflow, the software developer simply enters the Ishu Key in its Commit to run the Isho Trigger. There are also plugins that require developers to enter keystrokes when customizing their code. With the help of these plugins, which are enabled and configured in the Hooks section of BitBact software, you do not have to constantly warn developers to enter the key related to the Ishos in their comment. By forcing users, it will become a habit for them after a while.

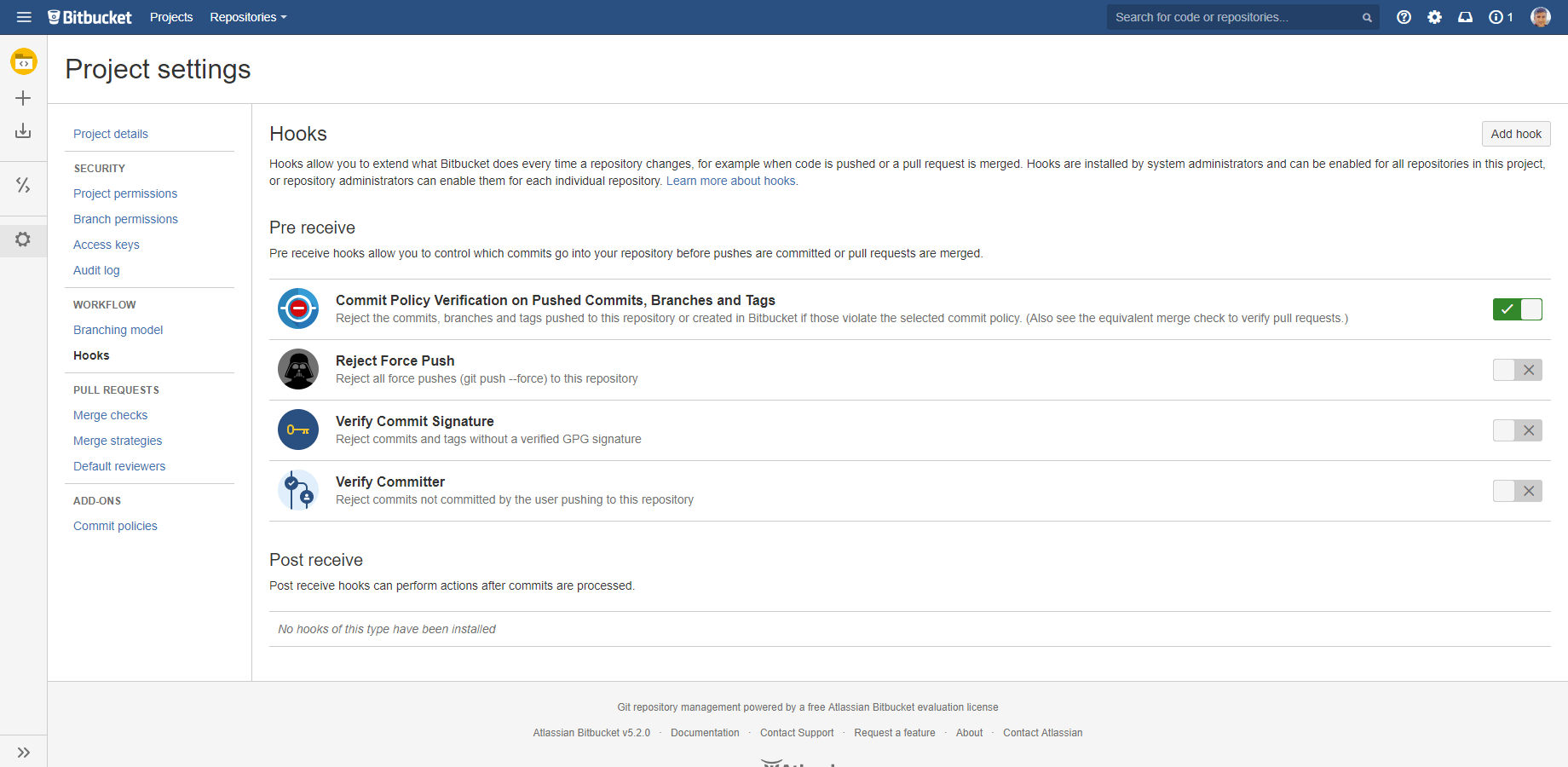
**Using Hooks in BitBact**

The Hooks feature in BitBact (server version) is a good way to automate common code management activities. Using Hooks provides a good way to manage the complexities of software development. To view the features provided by the Hooks section in BitBact, select the Hooks option from the Settings section and the workflow section. There are default items in this section.

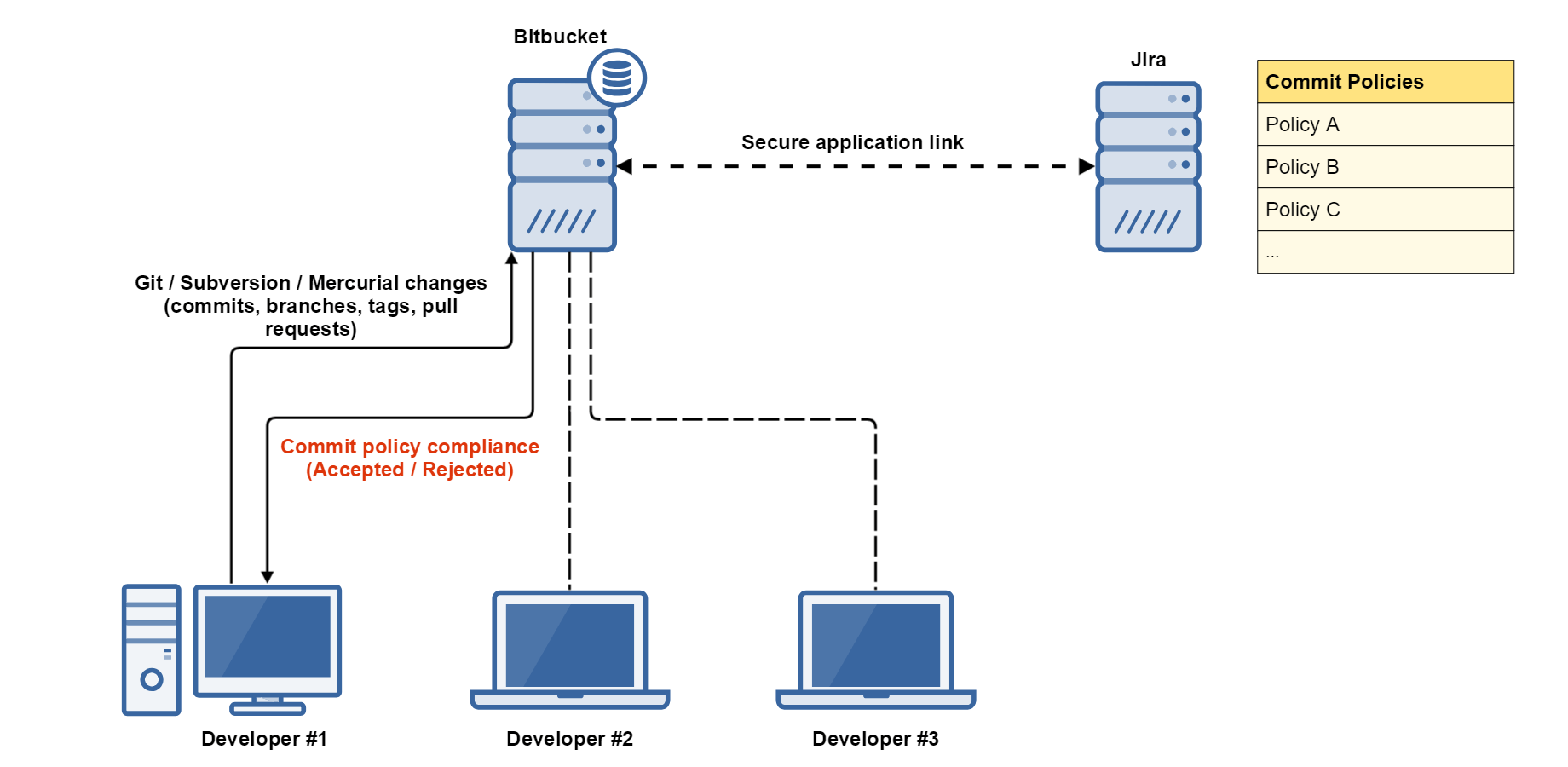


Integrating Jira and Bitbakt software will provide good options for using Hooks. There are hooks for scanning comments in the repository to automatically convert all TODO comments to one in Jira.

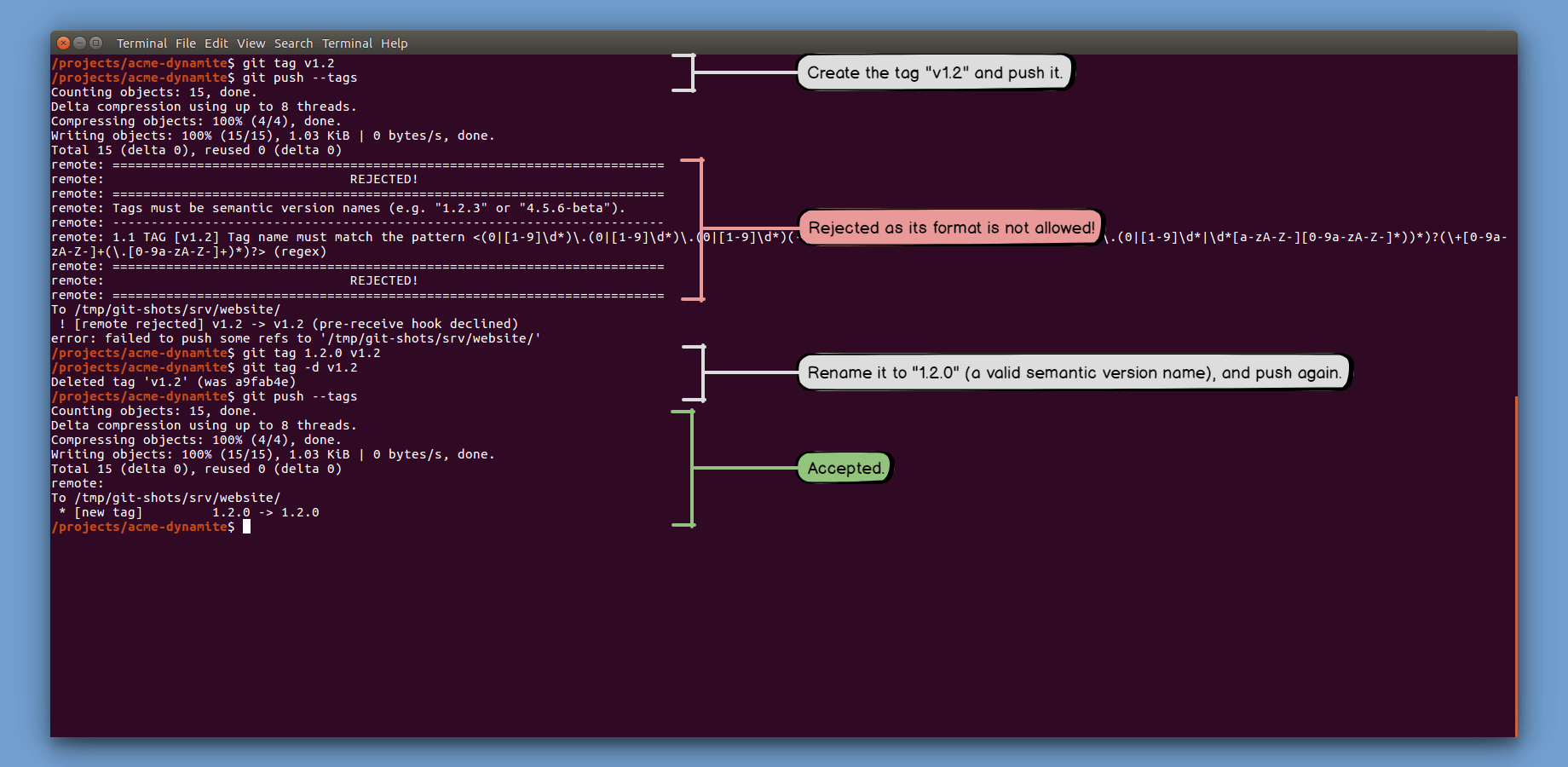
Related extensions can be used to extend the use of hooks. For example, by installing the [Better Commit Policy for Bitbucket](https://marketplace.atlassian.com/apps/1214430/better-commit-policy-for-bitbucket?tab=overview&hosting=server) plugin , you will be able to set appropriate policies for code customization by developers.



These policies and rules are set out in JIRA software, and when the developer intends to perform activities such as commit, branch, tag, pull request, Bitbakt will adapt the policies to JIRA and allow the developer to perform this activity if all is accepted. will be given. These policies will help to develop appropriate procedures among developers.



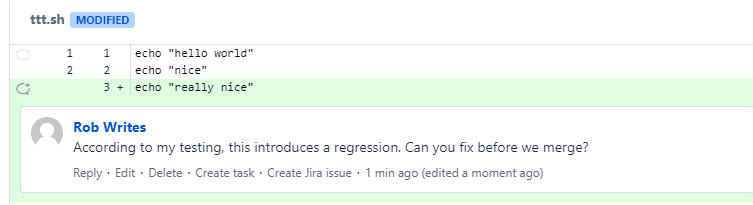
For example, if a developer does not use the correct tagging policies, they will receive an error and need to modify the tags according to the policies.



Making Jira's Ishos from the code review environment in Bitbact

One of the things that integrates well with Gera and BitBact is the work on pull requests.

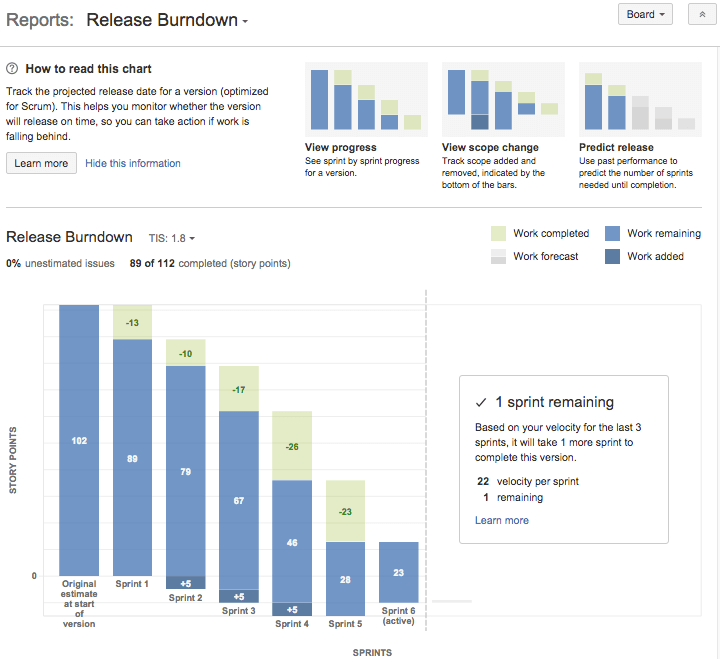
BitBack provides a great environment for users to view and review pull requests. Once Jira and BitBact are integrated, it will be possible for the user to perform the code task directly when the user is reviewing the code, without the need to exit. Create a code from the review environment and assign it to the relevant developer in JIRA.



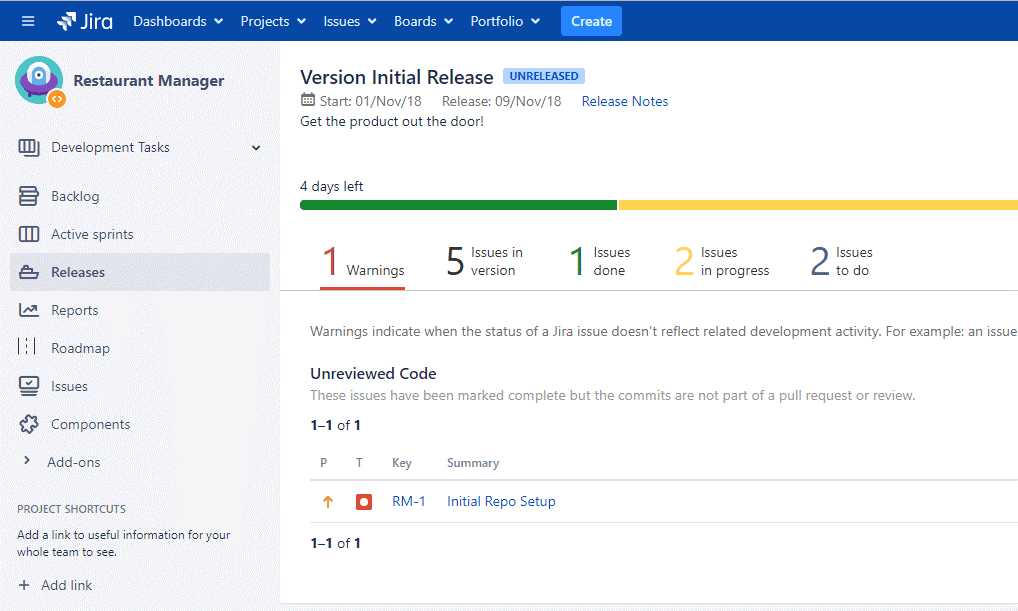
**Use Release Hub**

Another great tool in JIRA projects is the use of release management in this software, also known as Release Hub, and it is activated by integration with BitBact software.  
The Release Hub section allows you to visually view the status of issues related to scheduled releases.

This section provides great visual charts to see the status of tickets and works, until the release date.



The Release Hub section also allows you to see the status of the items you have selected for a Release. In this section, each Ishu will include two situations. Its status in the Jira workflow and its status in the related repository in BitBacket software.



**Result**

Jira and BitBact software is one of the most widely used software in the field of software project management and code management, both of which are Atlassian products. The ability to integrate the two software with each other and the features it provides allows the software development process to be designed as automatically as possible and to prevent common tasks and activities from being done manually by users. This allows developers to focus more on their work.