Google Summer Of Code 2016

Jenkins project : Support core plugin improvements

Project proposal

Minudika Malshan
Undergraduate
Department of Computer Science and Engineering
University of Moratuwa
Sri Lanka

Content

Introduction

Jenkins is an open source cross platform which provides continuous integration and delivery services for software development. It is a server-based system running in a servlet container such as Apache Tomcat.

Jenkins provides a rich extensibility through its plugins. Most of the parts of jenkins can be easily modified and improved. These features together with jenkins plugin echo system provides user a sophisticated environment for continuous integration in software development.

The Support-Core plugin provides the basic infrastructure for generating "bundles" of support information within Jenkins.
There are two kinds of bundles:

- Automatic bundles, these get saved in $JENKINS_HOME/support once per hour starting 15 seconds after Jenkins starts the plugin (i.e. this may still be generated even if Jenkins will not fully start). The automatic bundles are retained using an exponential aging strategy, so you should have a bunch of them over the entire lifetime once the plugin has been installed.
- On demand bundles, these are generated from the root "Support" action.

In this project scope, there are three features and improvements we are going to consider.

1. Ease the management bundles by the administrator (JENKINS-33090) [1]
2. Adding an option to anonymize customer **labels***. (JENKINS-33091) [2]
3. Allowing user to create an issue and submit a bundle into the OSS tracker using the support-core plugin. (JENKINS-21670) [3]

***** ***labels*** *: strings created by the user such as name of a job, folder, view, slave, and template*

Project Goals

At the end of the mid term, the following goals would be achieved.

- Under implementation related to JENKINS-33090,
    1. A UI component which lists all the generated bundles with the details such as date created, size, name and allows user to browse, delete existing bundle and create a new bundle.
    2. Implementation of above functionality at the backend.
    3. Test cases for the implementation

At the end of the project, the following goals would be achieved including the above deliverables.

4. In order to get JENKINS-21670 solved, the logical part of implementing listing of user created labels, creating randomized tokens, and producing a mapping will be done at the back end of support core plugin.
5. A UI component which allows user to open a ticket in jira using support core plugin UI with attaching generated bundles. (JENKINS-33091)

6. A proper documentation for the implemented features.
7. Automatic tests for implemented components and functionalities.

Implementation

The final deliverable of this project would be a improved version of support core plugin with the above mentioned features. Let’s consider those features one by one.

Implementation of ease the management bundles by the administrator.

In order to achieve this goal, I have to implement the functionalities for,

1. Listing bundles stored on the jenkins instance with their details. (name + creation date + size).
2. Allowing user to download each bundle.
3. Allowing user to delete each bundle or all bundles.
4. Allowing user to browse the content of each bundle.

Getting details of the created bundles can be done at the backend without much effort. Also backend implementation for deleting and creating bundles can be done.

In order to enable user to interact with the plugin, a suitable UI component should be implemented for above feature. For this, Apache Jelly can be used since it has been already used for the existing support core plugin UI.Figure 1 shows a sketch of this UI.

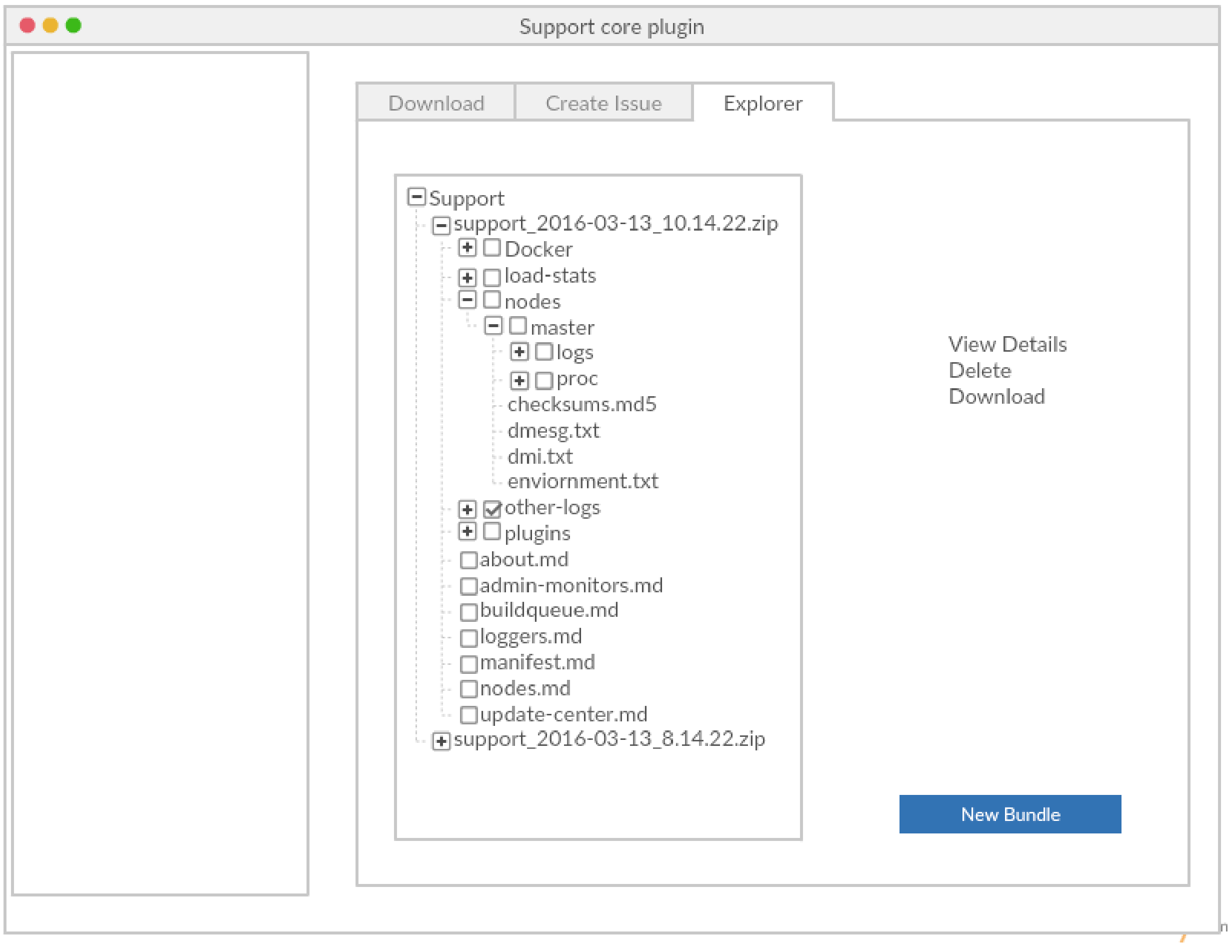


Figure 1 : Sketch of proposed UI for exploring/creating/deleting bundles

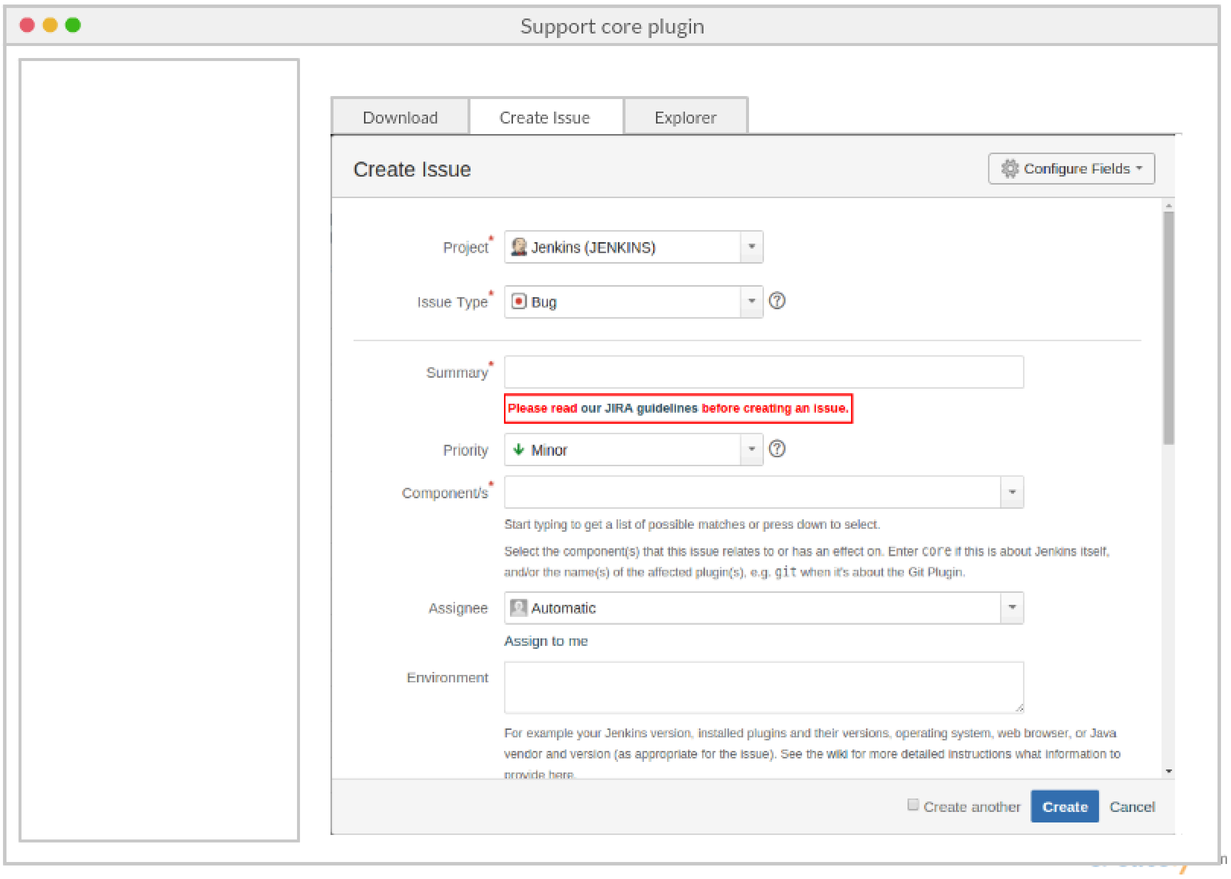
Enabling user to create an issue and submit a bundle into the OSS tracker

When a Jenkins user sees an issue , he/she commonly contacts his support contacts (Jenkins instance admins) and then Jenkins admins troubleshoot the issue.
I have to implement a feature which enables the user to report an issue to a admin through support core plugin.

Taking another step forward, it is possible to implement an option which enables user to create a ticket in jenkins issue tracker from the support core plugin. This requires a new UI component which gives the facility to create a ticket in the same way we create a ticket in the issue tracker directly.

In order to give these facilities to the user, we need to add an option to the plugin which helps the user to navigate between contacting with the admins or creating a new issue in issue tracker at once.

When it comes to create an issue from support core plugin, the UI component should be able to get details related to the issue which is going to be created and create and/or attach existing/new bundle to the ticket. Then support core plugin should be able to publish that ticket in the jenkins issue tracker.



Create Issue

Project* Jenkins (JENKINS)

Issue Type* Bug

Summary*

Please read our JIRA guidelines before creating an issue.

Priority Minor

Component/s*

Start typing to get a list of possible matches or press down to select.

Select the component(s) that this issue relates to or has an effect on. Enter `core` if this is about Jenkins itself, and/or the name(s) of the affected plugin(s), e.g. `git` when it's about the Git Plugin.

Assignee Automatic

Assign to me

Environment

For example your Jenkins version, installed plugins and their versions, operating system, web browser, or Java vendor and version (as appropriate for the issue). See the wiki for more detailed instructions what information to provide here.

Figure 2 : Sketch of proposed UI for creating issues from support core plugin

In order to publish tickets from support core plugin it requires jira credentials for the user. To manage these credentials, I hope it would be possible to use jenkins credentials plugin. [7]

This implementations can be done in two ways.

1. Jenkins is using Atlassian JIRA Project Management Software for its issue tracker. JIRA provides a REST API [4] and a JAVA API [5] which can be used for this implementation. Publishing the newly created issue on the issue tracker can be invoked through REST calls.
2. Jenkins already has a jira plugin.[8] It would be possible to integrate its feature with the support plugin.

When we create bundles to attach with the ticket, it is important to protect user's privacy who creates the ticket. When considering doing that, anonymizing user created labels (texts) comes to the front.

Implementing an option to anonymize customer labels

In order to implement this option, I will have to implement three functionalities.

1. Creating randomized tokens for labels created by users.
2. Produce a mapping for those labels.
3. Substituting encoded labels into all the files included in the support bundle.

These three functionalities will be implemented at backend and used when user creates a ticket in issue tracker using support bundle.
When creating randomized tokens, it would be much useful and effective if we can create those tokens in a way they make sense to humans. (i.e. readable to humans). For that, I am hoping to use a suitable java library to create human friendly random tokens. One of such libraries is "wordnet-random-name". [6]

However to substitute randomized tokens, all files included in the bundle should be read. This can become inefficient when bundle consists of large number of files. I am planning to do some research and optimize this file read/write operations as much as possible.

Timeline

April 22 - May 22:

- Getting familiar with,
    - the source code of support core plugin
    - Apache Jelly
    - Jira's REST API
    - Jenkins JIRA plugin
    - Jenkins credentials plugin

- Doing research on
    - optimizing file read/write operations
    - 3rd party libraries which can be used for the project and the compatibility of their license.

May 22 - June 12

- Implementation of frontend and backend related to JENKINS-33090
- Testing
- Doing optimizations and bug fixes.
- Create a documentation for implemented feature.

June 12 - July 10

- Implementation of frontend and backend related to JENKINS-33091
- Writing test cases.
- Testing and Fixing bugs.
- Create a documentation for implemented feature.

July 10 - August 10

- Implementation of frontend and backend related to JENKINS-33091
- Writing test cases
- Testing Bug fixing

August 10 - August 16

- Testing and verifying overall implementation.
- Finalizing the documentations.
- Submitting the source code and the documentations.

Possible challenges

When considering challenges I have to face during the project, Using Jelly is the newest thing I have to get familiar with. Getting familiar with the rest of the source code, which is I am currently doing would not be a big issue.

And the license of the 3rd party libraries which will be used for this project should be compatible with the MIT license under which Jenkins is released.

Future Contribution

I am willing to work on further improvements which would be required for the features implemented under this project. And also I am happy to contribute on jenkins development as well.

About Me

I am Minudika Malshan, an undergraduate student in Computer Science and Engineering from University of Moratuwa, Sri Lanka.In the field of Computer Science.
I have a good theoretical and practical knowledge in Java, Python, Html, JS , Git ,software engineering concepts and best practices.I am willing to work with and adaptable to new technologies as well.

As a person who is passionate in open source software development and seeking for new knowledge and experience , I am willing to give my contribution for this project.

Name	Minudika Malshan Gammanpila
Email	minudika001@gmail.com
Postal Address	292/8, 2nd lane, Badulla Road, Bandarawela
LinkedIn	https://lk.linkedin.com/in/minudika
GitHub	https://github.com/minudika

References

[1] https://issues.jenkins-ci.org/browse/JENKINS-33090
[2] https://issues.jenkins-ci.org/browse/JENKINS-33091
[3] https://issues.jenkins-ci.org/browse/JENKINS-21670
[4]
https://developer.atlassian.com/jiradev/jira-platform/guides/other/guide-jira-remote-issue-links/jira-rest-api-for-remote-issue-links

[5]
https://docs.atlassian.com/jira/5.0-beta3/com/atlassian/jira/bc/issue/link/RemoteIssueLinkServic
e.html
[6] https://github.com/kohsuke/wordnet-random-name
[7] https://wiki.jenkins-ci.org/display/JENKINS/credentials+Plugin
[8] https://wiki.jenkins-ci.org/display/JENKINS/JIRA+Plugin