Assignment 5

Continuous build system

IMT 3501 Software Security

Jan Kerkenhoff
(141628)
David Vierheilig
(140155)
Urs Oberdorf
(141627)
Christian Neßlinger
(140153)

Task

1. Sketch a continuous build system using a private/hybrid cloud for HiG/IMT. Consider the following functional requirements:
a) Students submit code to a common repository
b) Build system compiles C, C++, C#, Java source to binaries
c) Teachers retrieve source code+binaries
d) Deployment to web servers and app stores
e) Automated functional testing

2. What are the legal, organisational, and technical security requirements in software integration?
a) Requirements of students?
b) Requirements of teachers?
c) Requirements of system administrators?
d) Requirements of the institution?

Introduction

We were asked to sketch a continues build system for HIG, so that students could commit their code to a common repository, where the code would then automatically compile into binaries. After compiling the teacher could retrieve the binaries and source code and review it before they deploy it to webservers and app stores. The system should also be responsible for automatic functional testing.

In the following we will present our sketch for this case.

What is continues integration

In the last couple of years different approaches to software engineering have been published. The most common of this approaches nowadays is continues integration or CI. It is part of an agile development work flow that tries to minimize integration errors. When working with multiple people on the same part of a software it often occurs that differences between understandings, conventions and personal preferences cause problems when work needs to be merged back together. Therefor a lot of time is spent to fix these mistakes, which are sometimes even not reversible. Continues integration tries to solve these problem by building the software after each commit to the version control system. Therefore problems can be detected early on and fixed or avoid with the least effort required.

Requirements of the institution

1. The institution needs an internet connection which could handle 500 simultaneous connections.
2. The Server should have at least 16GB EEC RAM + 2 x Intel Xeon E7
3. The software should have the possibility for a detailed access management comparable with the access management from Active Directory.
4. The system should provide daily backups from the whole system to an other physical location.
5. The backups should available for up to 14 days.
6. The system should be a redundant system, minimum mirroring of the whole server.
7. The systems should be connected to an online uninterrupted power supply, to prevent hardware damages and data loss.
8. The system should be located in a temperature controlled environment.
9. The system location should be fireproofed.
10. Access to the system location should be controlled by an authentication system.

11. The system needs a qualified system administrator.

12. The data should be stored encrypted in all cases.

Requirements of system administrators

13. All connections between the system and the user should be encrypted with at least TLS.

14. The system should dynamically scale with the amount of users.

15. We will use a private cloud, hosted in-house at HIG.

16. The system should provide an up time of more than 99%.

17. Reaction time of system administrators should be less than 5h in case of an alarm from the monitoring system.

18. The system must include a monitoring with alarm capabilities.

19. There needs to exist a valid disaster recovery plan.

Requirements of teachers

20. Provided compiling support at least for C, C++ ,C# and Java.

21. The system has to offer the ability to comment and evaluate the code in the repositories.

22. Version control has to be provided.

23. Different software events must provide email notification to the respective users.

24. A dynamic issue tracking system must be present.

Requirements of students

25. Provide connections from outside the network.

26. System and client independent access to the system.

27. Localization support in english.

Solution

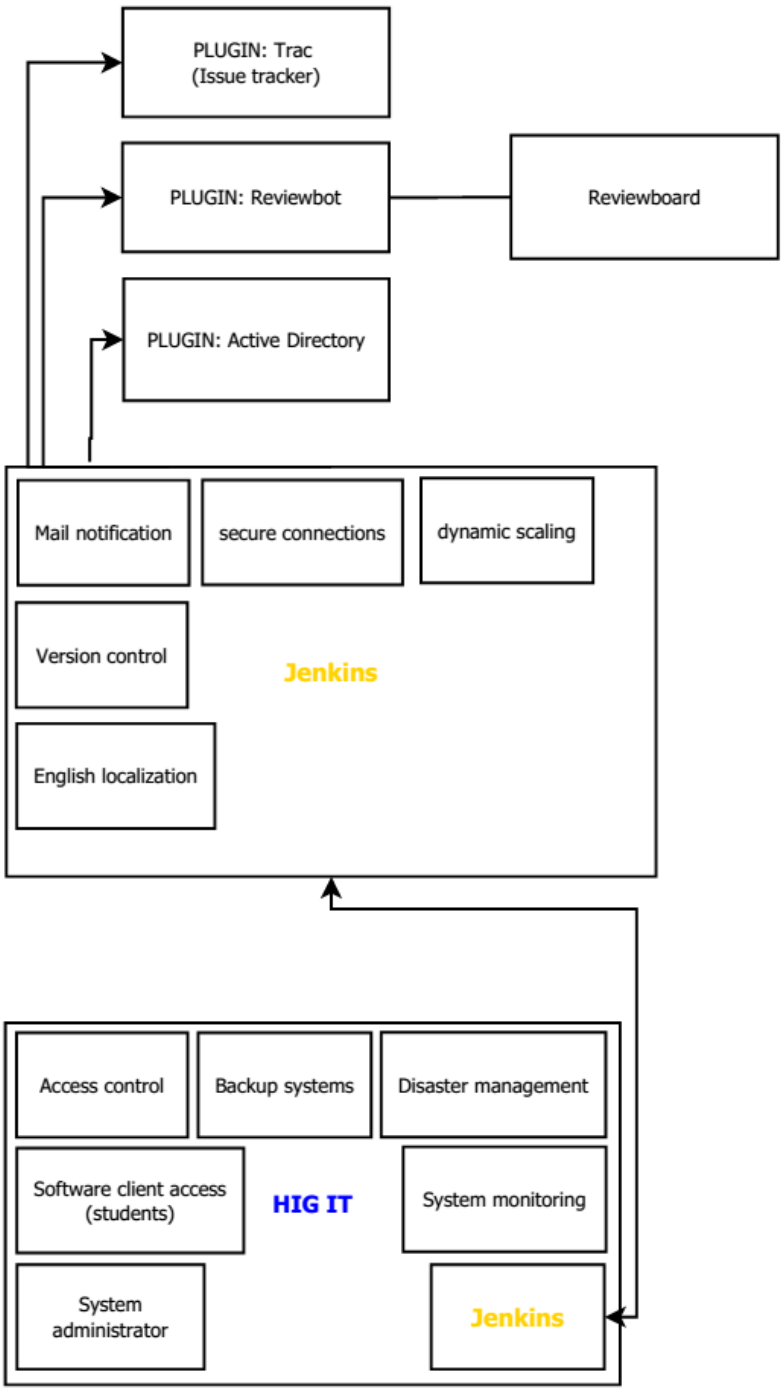
We decided to use Jenkins as a continues build system. Jenkins provides solution for the following requirements:

- Jenkins Active Directory plugin for the requirement three.
- Jenkins supports secure connections
- Jenkins supports dynamic scale using master and slave principle
- Jenkins supports the minimal requirement for the programming languages.
- Combination of Jenkins and Reviewboard fullfill the requirement 21.
- Jenkins supports Version Control
- Jenkins supports e-mail notification for different events.
- In Jenkins an issue tracking can be implemented with help of “trac”
- Jenkins supports the language in english.

We use Reviewboard for offering review on checkedin source code through teachers and students

We assume that the following requirements are satisfied by the already established infrastructure of HIG IT Services:

Requirements one to twelve and fourteen to nineteen.



Conclusion

We have chosen Jenkins because it fulfills nearly all of our requirements and with the help of Jenkins plugins like Active Directory every requirement is covered. Another reason for choosing Jenkins is that it's open-source, which fulfills the open design postulate from Saltzer and Schroeder.

We didn't separate the tasks because we worked the whole time together.

Links:
Trac plugin:
https://wiki.jenkins-ci.org/display/JENKINS/Trac+Plugin
Reviewboard:
https://www.reviewboard.org/
Reviewbot (plugin for the Reviewboard):
https://wiki.jenkinsci.org/display/JENKINS/Jenkins-Reviewbot
Jenkins LTS:
http://jenkins-ci.org/
Trac:
http://trac.edgewall.org/
Saltzer and Schroeder:
http://emergentchaos.com/the-security-principles-of-saltzer-and-schroeder