International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 2 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT1952338

HLR Framework Development for Continuous Integration

Chaitra N. Korlahalli[1], Prof. Sandhya S. [2]

[1]PG Scholar, Computer Network Engineering, R V College of Engineering, Bangalore, Karnataka, India
[2]Assistant Professor, Department of Computer Science, R V College of Engineering, Bangalore, Karnataka, India

ABSTRACT

Home Location Register (HLR) and Home Subscriber Server (HSS) are the master databases which manages Second Generation (2G), Third Generation (3G) and Fourth Generation (4G) networks. The Continuous Integration (CI) process requires engineers to integrate application features into a single repository frequently. In this paper, an automated HLR framework development for testing the 2G, 3G, 4G or LTE and IMS application features is presented. The HLR framework helps to install and configure the HLR test machines. This framework helps to automate the configuration management on the test machines using Jenkins server to run Sanity and Regression. The HLR Framework Development for Continuous Integration is going to reduce the manual efforts by automating the task of Continuous Integration process such as installation, configuration, execution and generation of a report.

Keywords : Home Location Register (HLR), Front End(FE), Continuous Integration (CI), Sanity, Regression.

I. INTRODUCTION

Continuous integration (CI) is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily, leading to multiple integrations per day. Using automated build (including test), each integration is verified to detect integration errors as quickly as possible. Home Location Register (HLR) is the master databases which manage Second Generation (2G), Third Generation (3G) and Fourth Generation (4G) networks. It supports in a flexible way and provides Authentication, Authorization and Accounting (AAA) for subscriber data. HSS binds HLR functions for 2G and 3G networks and HSS functions for 4G and Internet Protocol Subsystem (IMS) networks into a single solution. 2G is based on Global System for Mobile communication, this provides various services such as picture messages text messages, multimedia messages and voice calls. 3G provides video calls, conference calls and faster data transfer. 4G provides more consistent mobile broadband internet for smartphones, tablets, laptops and it provides more internet speed compared to 3G. IMS provides voice calls over Internet Protocol (IP) using packet-switched network rather than the traditional circuit-switched network. Packet-switched network data is transferred by dividing the data into separate packets and each packet takes a different route through the network to reach destination host and in circuit-switched network fixed route is established before communication starts between the nodes.

In this paper, we present our experiences in implementing & using the proposed framework. Our framework provides multiple benefits by automating repetitive and error-prone processes. This framework helps to automate the configuration management on the test machines using Jenkins server to run Sanity and Regression. Sanity testing checks whether all

software functionalities work in a proper way. Regression testing helps to ensure all previously developed and tested software still perform as expected when integrated with the latest software. Automated build helps in finding any product issues in the application, generate the automatic test report and send it to respective test suite owners to give a rapid feedback.

The organization of this document is as follows. In Section 2, the methodology used to develop the framework is explained. In section 3, the advantages is mention. In section 4, the expected results of the framework and its outcome is explained. In section 5, the conclusion and the future work related to the framework is explained. Section 6, it explains the future work that can be done to enhance the develop the framework.

II. METHODOLOGY

HLR Framework Development for Continuous Integration is being developed using technologies such as Python and Shell Script.

Shell script : Shell script is used to automate the HLR machines installation and configuration by performing remote login using ssh (secure shell) command and perform sequential execution of commands in different machines. A shell script sed command is used to perform insert, delete, replace or substitute the files.

Python : In this framework python shutil, xlrd, xlwt, xlutils modules are used to automate the DUSpec TPD and Common TPD sheets. The pexpect module is used to login HLR machines and performs the required operations such as copying the latest script to HLR FE, macro replacements for each test suites and run the SoftAuc configuration script.

oneNDS: One NDS is a hierarchical database which stores all subscriber data which can be a range in 100 thousand to 250 million and provides that data for common and consistent view based on the request of HLR FE. To perform the Regression Testing One NDS.

This project provides the following main functionalities:

- Installation of Cloud test machines.
- Running post configuration.
- Running Sanity and Regression.
- Report generation.

III. ADVANTAGES

The aim of the project is to automate the test machine installation, Configuration, Sanity test, Regression and generation and publishing test report . After installation, the machine must be able to take the load of Diameter messages. The framework development is to provide a fully developed machine for the testers to run and verify calls. The machine must provide support for regression and sanity runs. It must automatically connect to the Jenkins server which generates logs and reports for the calls which are executed. With the help of the framework, the manual effort will be reduced and stability of the systems can be maintained.

IV. RESULTS AND DISCUSSION

The main functionality of the developed framework is to construct the common platform, upon which the test machines are deployed. The post configuration is done. Then sanity and regression test is run to check the proper working and configuration of the deployed machines. The final result is generation of the regression report, analyse it and publish it in the confluence page for future analysis by the developers. The following snapshots shows the expected outcome of the framework.

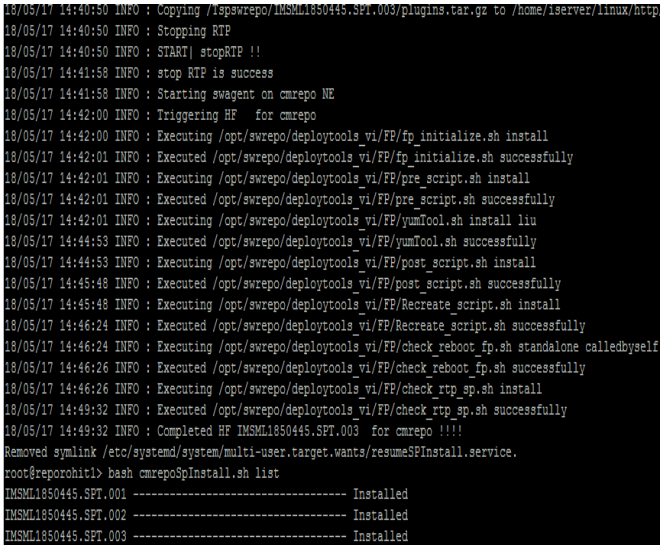

Fig 1: The HLR FE platform installation details

After installing platform installation, HLR Front End (FE) will be installed on top of platform. Fig 1 shows the status of HLR FE platform installation details.

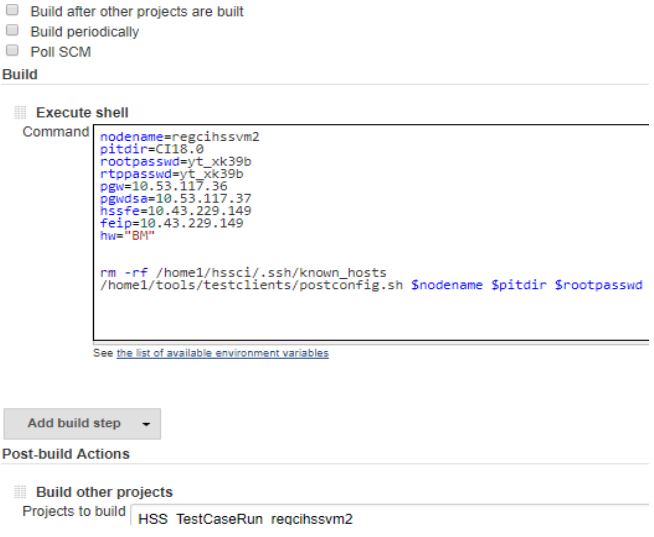

Fig 2 : Post Configuration details in Jenkins

The Fig 2 shows the details of configuration job in Jenkins server.

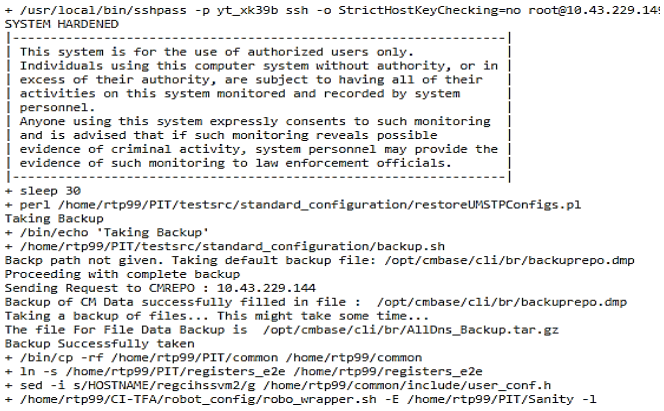

Fig 3: Post Configuration running details

Once the configuration job triggered in Jenkins server it will login to all HLR FE machines and do certain post configuration job. The Fig 3 shows the configuration running details in Jenkins server.

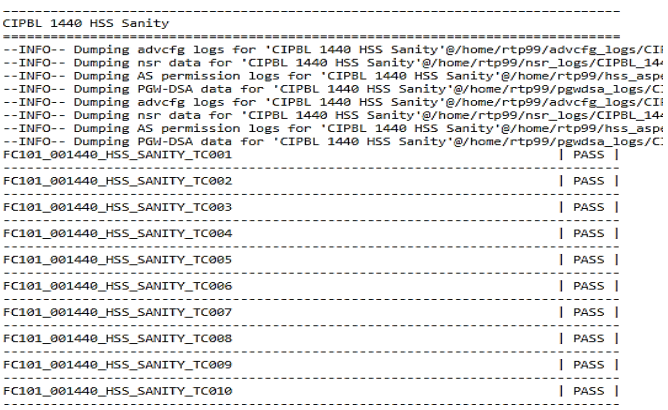

Fig 4 : Sanity running details

To verify the HLR machines are configured successfully or not, the Sanity test cases will be run. Fig 4 shows the running Sanity test case details.

```
ldap_add: Already exists (68)
Tsv Name: CIPBL-1424_Suite is getting executed using below command
pybot.bat -d CIPBL-1424_Suite -o CIPBL-1424_Suite.output.xml -l CIPBL-1424_Suite.l
C:/GIT/registers_e2e/ScriptsHSS/17.5/CIPBL_1424_LegacyA3/CIPBL-1424_Suite.tsv

==============================================================================
CIPBL-1424 Suite
==============================================================================
FC101_001424_HSS_TC001                                                | PASS |
------------------------------------------------------------------------------
FC101_001424_HSS_TC005                                                | PASS |
------------------------------------------------------------------------------
FC101_001424_HSS_TC007                                                | PASS |
------------------------------------------------------------------------------
FC101_001424_HSS_TC007_2                                              | PASS |
------------------------------------------------------------------------------
FC101_001424_HSS_TC020                                                | PASS |
------------------------------------------------------------------------------
FC101_001424_HSS_TC021                                                | PASS |
------------------------------------------------------------------------------
FC101_001424_HSS_TC022                                                | PASS |
------------------------------------------------------------------------------
FC101_001424_HSS_TC023                                                | PASS |
------------------------------------------------------------------------------
FC101_001424_HSS_TC024                                                | PASS |
------------------------------------------------------------------------------
FC101_001424_HSS_TC025                                                | PASS |
```

Fig 5: Regression running details

After all the Sanity test cases are executed successfully, the Regression test suites will run. Fig 5 shows the running details of Regression.

CI Report

Consolidated report

Summary

Total	Passed	Failed	Time	Percentage
4521	4028	493	0:31:32	89.1

regcihssvm2:

Total	Passed	Failed	Time	Percentage
4521	4028	493	0:31:32	89.1

Fig 6 : Regression report

Once the Regression run is over, the report will be generated and sent to respective CI Engineer. Fig 6 is a sample report of Regression test suites.

V. CONCLUSION

The purpose of HLR Framework Development for Continuous Integration is to reduce the work of CI engineers to perform test machine Installation, Configuration, Sanity test, Regression run and generation of test report through automation which has been completely successfully. This framework has drastically reduced the manual effort and risk of human errors while performing the above actions. This application helps in effective and efficient use of the resources.

VI. FUTURE WORK

HLR Framework is developed to reduce the manual effort of Continuous Integration team. This framework is made according to the specification but some following enhancements can be implemented in this project:

- The framework can be enhanced to troubleshoot the errors found during the installation or configuration.
- The framework can be enhanced to raise automation JIRA when any test suites are failing continuously.
- The framework can be enhanced to validate multiple test suites parallelly.
- The framework can be completely automated without any human interface.

VII. REFERENCES

[1]. Kim, Eun Ha, Jong Chae Na, and Seok Moon Ryoo. "Test automation framework for implementing continuous integration." Information Technology: New Generations, 2009. ITNG'09. Sixth International Conference on. IEEE, 2009.

[2]. Soni, Mitesh. "End to end automation on cloud with build pipeline: the case for DevOps in insurance industry, continuous integration, continuous testing, and continuous delivery." Cloud Computing in Emerging Markets (CCEM), 2015 IEEE International Conference on. IEEE, 2015.

[3]. Kim, Eun Ha, Jong Chae Na, and Seok Moon Ryoo. "Implementing an effective test automation framework." Computer Software

and Applications Conference, 2009. COMPSAC'09. 33rd Annual IEEE International. Vol. 2. IEEE, 2009

[4]. Shepard, Michael. Getting Started with PowerShell. Packt Publishing Ltd, 2015.

[5]. Chun, Wesley J. Core Python Applications Programming. Prentice Hall Press, 2012.

Cite this article as :

Chaitra N. Korlahalli, Prof. Sandhya S., "Home Location Register (HLR), Front End(FE), Continuous Integration (CI), Sanity, Regression", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5 Issue 2, pp. 1276-1280, March-April 2019. Available at doi : https://doi.org/10.32628/CSEIT1952338
Journal URL : http://ijsrcseit.com/CSEIT1952338