**ACKNOWLEDGEMENT**

I would like to express our profound grateful to Management of National Institute of Engineering, Mysuru for providing me an opportunity to complete my academics in this esteemed institution.

I express my gratitude to **Dr. G. L. Shekar,** Principal, National Institute of Engineering, for providing me an excellent facilities and academic ambience; which have helped me in satisfactory completion of project work.

I extend my sincere thanks to **Mr. Girish**, Head of the Department, Master of Computer Applications; for providing me an invaluable support throughout the period of our project work.

I wish to express my heartfelt gratitude to **Mr. Ananth .G.S,** Assistant Professor, Dept of Master of Computer Applications for his valuable guidance, suggestions and cheerful encouragement during the entire period of our project work.

I express my truthful thanks to **Mr. Sanjay,** Assistant Professor, Project Coordinator, Dept of Master of Computer Applications for his valuable support.

Finally, I take this opportunity to extend my earnest gratitude and respect to our parents and family members, Teaching & Non-teaching staffs of the department, the library staff and all my friends, who have directly or indirectly supported me during the period of my Project work.

Regards,

**Vidyashankar Giri**

**4NI14MCA55**

**ABSTRACT**

Delivery of software solutions at a faster pace is essential in the industry today and to do that, the concept of automation came up and the IT industry has tried to implement the concept of automation to fasten the software production process and to speed up the process of product delivery to customer.

The proposed system does all the operations related to the software development and delivery process in an automated way. DEVOPS model has been incorporated in the system to carry out the task. Initial phase will carry out the manual installation and configuration of all the tools and services required for the proposed system and finally all the manual installations required for the release and deployment are automated using an automation tool called puppet.

The system is built by using tools like Git (Repository), Jfrog, Jenkins, Jira, Maven, Cobertura, Nagios (Monitoring tool) and puppet. These tools cover all the aspects of software development and deployment. The proposed system automatically builds the code, tests the code, bundles the packages, deploys the application and monitors the overall system.

TABLE OF CONTENTS

ACKNOWLEDGEMENT i

ABSTRACT ii

LIST OF FIGURES iv

LIST OF SYMBOLS AND ABBREVIATIONS vi

COMPANY PROFILE vii

1 INTRODUCTION 1

1.1 INTRODUCTION TO AUTOMATION 1

1.2 PROBLEM STATEMENT 2

1.3 OBJECTIVES OF THE PROPOSED SYSTEM 3

1.4 SCOPE OF THE SYSTEM 3

1.4.1 CONTINUOS INTEGRATION 3

1.4.2 CONTINUOS DEPLOYMENT 3

1.4.3 CONTINUOS TESTING 3

1.4 CONCEPT OF DEVOPS 4

1.5 DEVOPS CHAIN 5

1.6 SITE RELIABILITY ENGINEERING 6

1.7 INTRODUCTION TO AGILE METHODOLOGY 7

1.8 OVERVIEW OF THE PROJECT 8

1.9 ORGANISATION OF THE THESIS 8

2 LITERATURE SURVEY 9

2.1 EXISTING SYSTEM 9

2.2 LIMITATIONS OF EXISTING SYSTEM 10

2.3 PROPOSED SYSTEM 10

2.4 DRAWBACKS OVERCOME BY PROPOSED SYSTEM 12

2.5 TOOLS USED 12

3SYSTEM REQUIREMENT SPECIFICATIONS 15

3.1 SYSTEM REQUIREMENT 15

3.2 FUNCTIONAL REQUIREMENT 15

3.2 NON-FUNCTIONAL REQUIREMENT 16

3.2.1 AVAILABILITY 16

3.2.2 CAPACITY 17

3.2.3 CONTINUITY 17

3.2.4 SECURITY 18 3.3 HARDWARE REQUIREMENT 18

3.4 SOFTWARE REQUIREMENT 18

4 SYSTEM ANALYSIS 19

4.1 PROJECT DESCRIPTION 19

4.1.1 CONTINUOS INTEGRATION 19

4.1.2 CONTINUOS TESTING 20

4.1.3 CONTINUOS DELIVERY 21

4.1.4 CONTINUOS DEPLOYMENT 22

4.2 HIGH-LEVEL DESIGN 23

4.3 FEASIBILITY STUDY 24

4.3.1 TECHNICAL FEASIBILITY 25

4.3.2 ECONOMIC FEASIBILITY 25

4.3.3 OPERATIONAL FEASIBILITY 25

4.3.4 MOTIVATIONAL FEASIBILITY 25

4.3.5 SCHEDULE FEASIBILITY 26

4.3.6 FEASIBILITY CHECK 26

5 SYSTEM DESIGN 27

5.1 SYSTEM ARCHITECTURE 27

5.1.1 AUTOMATED DEPLOYMENT AND MONITORING 27

5.1.2 DESCRIPTION OF ARCHITECTURE 27

5.2 DATAFLOW DIAGRAM 29

5.3 USECASE DIAGRAM 30

5.4 SEQUENCE DIAGRAM 31

5.5 ALGORITHMS 32

5.5.1 ALGORITHM TO MONITOR USED SPACE 32

5.5.2 ALGORITHM TO MONITOR NO OF PROCESSES 33

5.5.3 ALGORITHM TO MONITOR PUPPET MASTER 34

5.5.4 ALGORITHM TO MONITOR JENKINS 34

5.5.5 ALGORITHM TO MONITOR XINETD 35

5.5.6 ALGORITHM TO AUTOMATE JAVA INSTALLATION 36

6 SYSTEM IMPLEMENTATION 37

6.1 PROJECT MANAGEMENT 37

6.1.1 JIRA 37

6.1.2 FEATURES OF JIRA 38

6.1.3 STRUCURE OF JIRA 39

6.1.4 JIRA WORK-FLOW 39

6.1.5 JIRA ISSUE CREATION 40

6.1.6 JIRA QUERY LANGUAGE 42

6.2 SOURCE CODE MANAGER 43

6.3 GIT 43

6.3.1 ADVANTAGES OF GIT 43

6.3.2 OPERATIONS IN GIT 44

6.3.3 GITLABS 46

6.3.4 WORKING WITH GITLABS 46

6.4 PROJECT BUILDING 46

6.5 MAVEN 47

6.5.1 MAVEN OBJECTIVE 48

6.5.2 MAVEN POM.XML PAGE 49

6.5.3 ELEMENTS OF POM.XML PAGE 49

6.5.4 MAVEN REPOSITORY 49

6.6 INTEGRATION SERVER 50

6.6.1 CONTINUOUS INTEGRATION SERVER 50

6.7 JENKINS 50

6.7.1 MANAGE PLUGINS 51

6.7.2 GIT INTEGRATION 52

6.7.3 MAVEN INTEGRATION 53

6.7.4 MAVEN SETTINGS 54

6.8 CONTINUOUS DEPLOYMENT SERVER 54

6.9 JFROG ARTIFACTORY 55

6.9.1 INTEGRATING WITH JENKINS 60

6.10 MONITORING TOOL 63

6.11 NAGIOS 63

6.11.1 INSTALLATION OF HTTPD 64

6.11.2 INSTALLATION OF NAGIOS 64

6.11.3 INSTALLATION OF XINETD AND NRPE 66

6.11.4 CREATION OF CUSTOM SCRIPT 67

6.11.5 CONFIGURATION OF MASTER AND SLAVE 73

6.11.6 TERMINAL OUTPUT OF HOST AND SERVICES 76

6.12 INSTALLATION PUPPET 77

6.12.1 INSTALLATION OF PUPPET MASTER 77

6.12.2 INSTALLATION OF PUPPET SLAVE 78

6.12.3 PUPPET SCRIPTS FOR AUTOMATION 80

7 SYSTEM TESTING 87

7.1 INTRODUCTION 87

7.2 DIFFERENT TYPES 87

7.2.1 UNIT TESTING 87

7.2.2 INTEGRATION TESTING 88

7.2.3 USER TESTING 88

7.3 SYSTEM TESTING 88

7.4 ACCEPTANCE TESTING 89

7.4.1 REGRESSION TESTING 89

7.4.2 STRESS TESTING 89

7.5 SCREENSHOTS

8 CONCLUSION AND FUTURE WORK 90

8.1 CONCLUSION 90

8.2 FUTURE WORK 90

9 BIBLIOGRAPHY 91

LIST OF FIGURES

1.1 DEVELOPMENT LIFE CYCLE 4

4.1 HIGH LEVEL DESIGN 23

4.2 LOAD BALANCER 24

5.1 AUTOMATED DEPLOYMENT AND MONITORING 27

5.2 DATA FLOW DIAGRAM 30

5.3 USE CASE DIAGRAM 31

5.4 SEQUENCE DIAGRAM 32

6.1 SCRUM BOARD 38

6.2 KANBAN BOARD 38

6.3 STRUCTURE OF JIRA 39

6.4 CREATE EPIC 40

6.5 CREATE STORIES 41

6.6 USER CREATION 42

6.7 GIT OPERATION 45

6.8 GIT LABS 46

6.9 MAVEN STRUCTURE 48

6.10 MAVEN POM.XML 48

6.11 JENKINS LOGIN PAGE 51

6.12 INSTALLING PLUGINS 52

6.13 INSTALLING TOOLS 52

6.14 FETCHING PROJECT FROM GIT 53

6.15 BUILDING PACKAGE 54

6.16 INSTALLING JFROG 55

6.17 ADDING JFROG USER 56

6.18 CHANGING OWNERSHIP 56

6.19 ADDING VARIABLES 57

6.20 INSTALLING JFROG SERVICES 57

6.21 JFROG INTERFACE 58

6.22 CREATING REPOSITORY 58

6.23 ARTIFACTORY DASHBOARD 59

6.24 CHANGING DATABASE TO MYSQL 59

6.25 INSTALLING ARTIFACTORY PLUGIN 60

6.26 ADDING ARTIFACTORY SERVER TO JENKINS 60

6.27 CREATING LOCAL REPOSITORY 61

6.28 ADDING TARGET REPOSITORY 61

6.29 BUILDING PROJECT TO ARTIFACTORY 62

6.30 ARTIFACTORY DEPLOYED 62

6.31 INSTALLATION OF HTTPD 64

6.32 INSTALLATION OF NAGIOS 65

6.33 INSTALLATION OF NAGIOS-2 66

6.34 INSTALLATION OF XINETD 66

6.35 INSTALLATION OF XINETD-2 66

6.36 INSTALLATION OF XINETD-3 67

6.37 LIST OF PLUGINS 67

6.38 PLUGIN TO CHECK SPACE USED 68

6.39 PLUGIN TO CHECK TOTAL RUNNING PROCESSES 69

6.40 PLUGIN TO CHECK TOTAL ACTIVE USERS 70

6.41 PLUGIN TO CHECK ARTIFACTORY STATUS 71

6.42 PLUGIN TO CHECK PUPPET MASTER STATUS 72

6.43 DEFINITION OF SERVER HOST TYPE 73

6.44 DEFINING COMMAND IN SLAVE 74

6.45 DEFINING COMMAND IN MASTER-1 75

6.46 CONSOLE OUTPUT OF HOSTS 76

6.47 CONSOLE OUTPUT OF SERVICES 76

6.49 DEFINING PUPPET MASTER 78

6.50 DEFINING PUPPET SLAVE 79

6.51 EXECUTION COMMAND OF SCRIPTS 80

6.52 AUTOMATED DEPLOYMENT SCRIPT 81

6.53 JFROG ARTIFACTORY 81

6.54 TOMCAT VIEW OF DEPLOYED WAR 82

6.55 AUTOMATED JAVA INSTALLATION 83

6.56 AUTOMATED JAVA INSTALLATION-2 83

6.57 AUTOMATED JENKINS INSTALLATION 84

6.58 GIT CLONE SCRIPT 85

6.59 GIT CLONE SCRIPT 85

6.60 SITE.PP FILE 86

LIST OF SYMBOLS AND ABBREVIATIONS

DevOps Development Operations

SRE Site Reliability Engineering

OPS Operations

HTTPD Hyper Text Transfer Protocol Daemon

XINETD Extended Internet Daemon

NRPE Nagios Remote Plugin Executor

ESXi Enterprise Server Xi

YUM Yellowdog Updater Modified

CI Continuous Integration

CD Continuous Deployment

RPM Redhat Package Manager

DNS Domain Name System

**COMPANY PROFILE**

****

Hewlett Packard Enterprise Company (commonly referred to as Hewlett Packard Enterprise or HPE) is an American multinational enterprise information technology company based in Palo Alto, California, founded on 1 November 2015 as part of splitting of the Hewlett-Packard company. HPES digital platform enables products across four distinct lines of businesses

* HPE SERVERS
* HPE NETWORKING
* HPE STORAGE
* HPE SERVICES

**HPE SERVERS**

In computing, a server is a computer program or a device that provides functionality for other programs or devices, called “clients”. This architecture is called the clientserver model, and a single overall computation is distributed across multiple processes or devices. Servers can provide various functionalities, often called “services”, such as sharing data or resources among multiple clients, or performing computation for a client.

**HPE NETWORKING**

HP Networking is the networking division of Hewlett Packard Enterprise, within the HP Enterprise Business segment. HP’s networking division was previously known as HP ProCurve. In April 2010, following HP’s acquisition of 3Com Corporation, HP combined the ProCurve and 3 Com entities as HP Networking.

**HPE STORAGE**

HP Enterprise Services is the business and technology services subsidiary of the Hewlett Packard Enterprise strategic business unit. It was formed by the combination of Hewlett-Packard’s legacy services consulting and outsourcing business and the integration of acquired Electronic Data Systems.

**HPE SERVICES**

HPE is one of the world’s leading providers of insurance software

**BIBLIOGRAPHY**

1. Hpe company documentations.
2. https://learn.perl.org/tutorials.
3. https://en.wikipedia.org/wiki/nagios.
4. www.puppetresources.com.
5. https://newrelic.com/devops/benefits-of-devops.
6. http://www.devopsdigest.com/devops-advantages-1.
7. www.slashroot.in/puppet-tutorial-introduction-puppet-configuration-management-tool.
8. https://www.nagios.com/solutions/network-monitoring-tools/.
9. https://docs.puppet.com/puppet/4.9/architecture.html.
10. Gorka Gallardo Josune Hernantes EEbert, Christof and Nicolas Serrano.
11. Devops. IEEE Software, vol.33, no.3:pp.94–100, 2016.
12. Matt Callanan and Alexandra Spillane. Devops: making it easy to do the right thing. IEEE Software, vol.33, no.3:pp.53–59, 2016.
13. Len Bass Zhu, Liming and George Champlin-Scharff. Devops and its practices. IEEE Software, vol.33, no.3:pp.32–34, 2016.
14. How to configure a single Jenkins job to make the release process from trunk or branches. http://stackoverflow.com/questions/ 17146202/configure-jenkins-to-work with-svn-branches.
15. Jenkins list git Tags and Branches.

https://gist.github.com/eeichinger/9761870.

1. Powershell and Continuous Integration (JENKINS). https://allthingssitecore.com/2014/10/20/powershell-and-continuous-intergration-jenkins/.
2. Post Build Scripting in Jenkins.

http://stackoverflow.com/questions/11160363/execute-shell-script-after-post-build-in-jenkins.

1. Software Architecture in DevOps.

https://dzone.com/articles/software-architecture-devops.

1. ”List Subversion Tags” - and branches and trunk.

https://groups.google.com/forum/#!topic/jenkinsci-users/4OpQ3BSSdyo.

1. Parameterizing the Builds in Jenkins. https://wiki.jenkinsci.org/display/JENKINS/Parameterized+Build.