International Journal of Scientific Research in Computer Science and Engineering
Vol.6, Issue.2, pp.87-95, April (2018)

Research Paper

E-ISSN: 2320-7639

Implementation of DevOps Architecture in the project development and deployment with help of tools

Chellamalla Mamatha[1*], S C V S L S Ravi Kiran[2]

[1]Dept. of CSE, CVR College of Engineering, Jawaharlal Nehru Technological University, Hyderabad, India
[2] Software Engineer, MAQ Software, Hyderabad, India

[*]*Corresponding Author: mamatha.reddy0546@gmail.com, Tel.: +91 99598 01353*

Available online at: www.isroset.org

Received: 10/April/2018, Revised: 18/April/2018, Accepted: 26/April/2018, Online: 30/April/2018

Abstract - DevOps is the portmanteau of development and operations. In recent times, the agile transformation was adopted in IT organizations for continuous integration principles in software development life cycle (SDLC) which has improved the efficiency of development in the project. With time being it has been realized that the optimization does not help in only continuous integration to make the software delivery process efficient. Unless all the modules in software delivery lifecycle are well designed, implemented and optimized. This is the problem with previous technologies and DevOps addresses it. This paper explains the various phases of SDLC, business needs and ways to move from continuous integration to continuous delivery.

Keywords - *DevOps, Continuous Development, Continuous Integration, Continuous Testing, Continuous Deployment, Continuous Delivery, Github, Jenkins, Maven, Ansible.*

I. INTRODUCTION

Patrick Debois, who devised the name "DevOps" in 2009 and he is also called as "The Father of DevOps" devised the name "DevOps" in 2009. The word DevOps says itself that it formed by coalescing the two words "Development" and "Operations". DevOps is the collaboration of development and deployment of software. DevOps is the portmanteau of development and operations. It is a software development method that escalates to the amalgamation between software development team and operations team. This is the time to change the old technology to new technology like DevOps. - "Time to stop wasting money, time to start delivering great software and building systems that scale and last" – Patrick Debois [15].

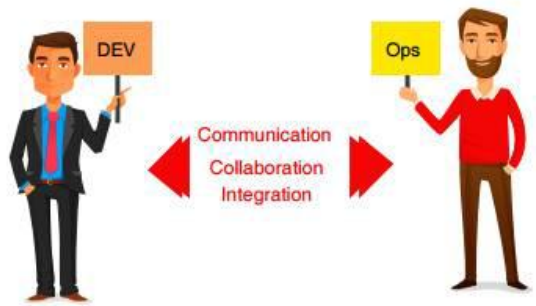

Figure 1: Understanding DevOps

The new software delivery procedure is adopted by the organizations since the market needs are changing continuously, rapid change in technology to deliver quickly. Customer waiting for 6 months or 1 year for a version to be released and giving feedback after release cannot happen nowadays. Customers need continuous engagement with the project so that they can provide the feedback continuously. In order to face the challenges, the organization should be lean and follow the agile transformation in all phases of SDLC. Over the years the organizations have adopted the agile transformation for optimization, but the evolution takes place to change the technology to DevOps [8]. It is important to keep all the phases in pace so that the software delivery lifecycle will not be delayed. DevOps is the mechanism which bridges the gap between developer-operations and not only limited to developer-operations but also for the continuous development, continuous testing and continuous integration [5]. DevOps main goal is to deliver the software rapidly with continuous development, continuous integration, continuous feedback and communication with development and operations team [1]. DevOps is the extension of agile principles in software delivery pipeline. DevOps principle plays an important role in complete SDLC, but it makes sense if both the development team and operations team work in a same pace [8].

Section I contains the introduction of DevOps - the collaboration of Development and Operations, Section II contains the related work of DevOps, Section III contain the Importance of DevOps, Section IV contains the Phases and delivery pipeline of DevOps, Section V contains the architecture of DevOps, Section VI describes results and discussion of DevOps and Section VII concludes research work with future directions).

II. RELATED WORK

A. WATERFALL MODEL

The first introduced Process Model is the Waterfall model. It is also referred to as a traditional model or linear-sequential life cycle model. It is very simple to understand and use. In this Waterfall model, every phase must be completed before going to the next phase and there is no issue of overlapping of phases. It is the first SDLC approach model for software development. The waterfall model illustrates the software development process flow in the linear sequential flow. The waterfall model process development is divided into separate phases and will start only after completion of the previous phase. The phases of Waterfall development model are as follows

1. Requirement gathering and Analysis
2. System Design
3. Implementation
4. Testing
5. Deployment
6. Maintenance

This model is the first model developed for the software development and ensures success, but the disadvantage is that each phase should be done only after completion of the previous phase it results in a long duration and also all the requirements are to be specified before starting the project [2].

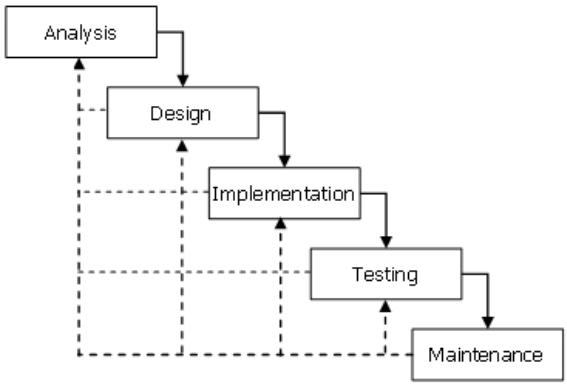

Figure 2: Waterfall Model

B. AGILE MODEL

Agile software development model is the process model which is the combination of both iterative and incremental model. The agile model focuses on process adaptability and customer satisfaction by rapid development and delivery of software product [12]. It breaks the complete product into small incremental builds and these builds are completed in iterations. These iterations last for one to three weeks and involve in cross-functional product development. The teams simultaneously work on these phases

1. Planning
2. Requirement Analysis
3. Design
4. Coding
5. Unit Testing
6. Acceptance Testing

This agile model after completion of system testing for the product it will be deployed in the market. It helps to complete the project rapidly. It is also suitable for static or dynamic requirements, but it does not suitable for handling complex dependencies and a high risk of sustaining, maintaining and extending [3], [4].

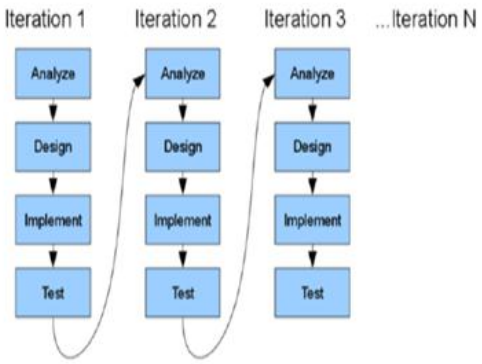

Figure 3: Agile Model

F

III. DEVOPS AND ITS IMPORTANCE

Patrick Debois, who is called as "The Father of DevOps" devised the name "DevOps" in 2009. DevOps (a portmanteau of development and operations) is a software development method that escalates to the amalgamation between software developers and information technology (IT) operation professionals.

A. DO WE REALLY NEED DEVOPS?

Developers always want to deliver the changes in the product as soon as possible whereas the operation team want reliability and stability in the product. This situation was explained clearly in "wall of confusion" by Lee Thomson is shown in figures 4 & 5. This wall of confusion not only gives the mentalities of two teams but also the tools they practice.

Development team uses some tools and Operations team uses different tools to perform the same task. DevOps bridge the gap between the development and operations for better and faster results.

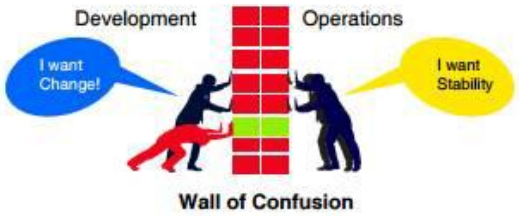

Figure 4: Wall of Confusion without DevOps

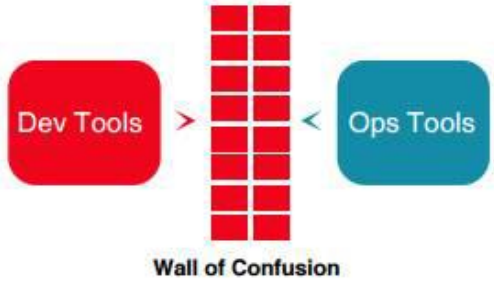

Figure 5: Wall of Confusion with DevOps

B. WHAT DRIVES THE NEED OF DEVOPS?

1. Strong collaboration between development and operation teams.
2. Synchronized deployment across multiple platforms.
3. Pressure to release applications to meet customer requirements or to enter into the new market.
4. Improving end user capability levels.
5. Vast usage of smart devices
6. Necessity to develop and deploy into cloud-based applications.
7. Increasingly complex IT infrastructure.
8. Need to reduce the cost for IT industry.

C. RECOGNIZING BUSINESS VALUE OF DEVOPS

DevOps applies agile and lean principles in the complete software deployment process to enhance the speed of delivery of product or service from the initial release to the production release and to the feedback given by the client based on the release. DevOps return our investment in these three areas

1. Enhanced Customer Experience

Delivering an enhanced product for the customer leads to build loyalty and increase in market share. To deliver an enhanced product we need to continuously obtain and respond to the customer's feedback faster and perform required changes suggested by the customer.

2. Increased capacity to Innovate

Lean thinking approaches are used in modern organizations to increase their innovation capacity. Their goals are to utilize the resources efficiently for other activities by reducing waste and rework. An example of lean thinking in organization is A-B testing in which large organizations asks a small group of users to test and rate two or more sets of software having different capabilities then the better capability set is picked up for the users and unsuccessful version is rolled back. This A-B testing is reliable only if efficient and automated mechanisms are adopted such as DevOps.

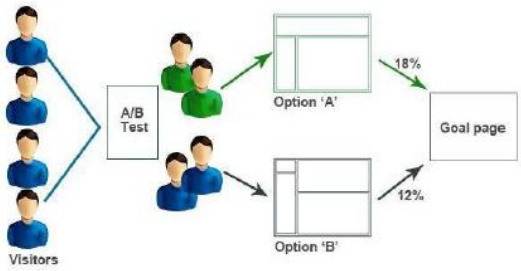

Figure 6: A-B Testing

A-B testing is the comparison of two web pages to know which will perform fast and efficient. It is also called as split testing. We compare two web pages by showing the two variants (let's call them A and B) to similar visitors at the same time. The one which has more conversion rate is accepted.

3. Faster time to value:

This involves in development of new culture and practices and automating the project leads to fast and reliable delivery process throughout the production phase. This DevOps can be worked as a business capability with the tools for release planning, predictability and success. The DevOps main goal is to deliver the value faster and in efficient way and the value definition changes with organization or with the project.

IV. DEVOPS PHASES AND DELIVERY PIPELINE

A. Continuous Planning

Business plans are already using agile methodologies to deliver quickly and change according to market conditions. It is better to have the checkpoints so that we can easily do the necessary changes given as feedback by customer. Dev / Test teams adapting to quick changes is not an easy task in business environments. DevOps allows us to prioritize the product backlogs and taking business perspective into consideration. This is the continuous process of planning, executing, getting feedback from the customer, the cycle continues [11].

B. Continuous Integration

Continuous Integration means dynamically integrating the changes made to the project to the team and not restricted to our local machine and validates the behavior of the code. Sharing with component teams but integrating beyond component boundaries at product integration level. Further the process optimization refers to automation as soon as the developer delivers the change build systems must detect the change and trigger a build taking sanity test and building repository. This must be a cyclic process across the development [9, 10].

C. Continuous Deployment

Continuous Deployment is the heart of the DevOps and acts as the Centre point to the complete software delivery optimization. Most of the surveys said that in many organizations the reason for the delay in software delivery is the operations. Hardware setting in the development build may vary from days to weeks. These deployment processes are inconsistent and manual. DevOps principles recommend the automation of deployment and hardware provision and cloud play a vital role in this field. DevOps proposes a concept called Infrastructure as a code (IAAC) which says that complete infrastructure provision should be maintained in source code repository [6].

D. Continuous Testing

Automation is the best option for continuous testing to test every test case. If any process need to do repeatedly for some constant time it is better to automate that process. They are humongous applications available in the market for do that type of testing process to meet the goal. There will be a chance of maximum to automate the manual testing process we need to evaluate on the same. Software delivery process must be able to execute the test suite automatically with the user intervention leading towards the goal reach easily. This kind of process not only makes testing process automate but also allows test cases to be carried out fast in production like system (deployment) [6].

E. Continuous Monitoring

As discussed in all the above approaches we adopted, there is a chance to observe various parameters and react to them accordingly. The capability to test early and production like systems we can react to them in timely manner.

DELIVERY PIPELINE

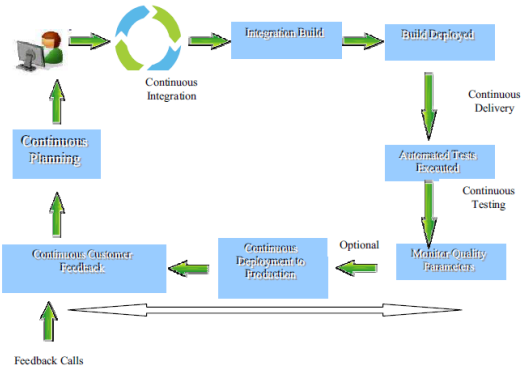

Figure 7: Delivery Pipeline

DevOps approach delivery pipeline is shown in above figure (figure 7). It can be compared with manufacturing unit delivery pipeline. Each build / release should undergo this cycle of dev - fvt - regression - stage - production - test phases by clearing all the quality parameters. With this automated pipeline there will be consistent releases [7, 8].

V. ARCHITECTURE OF PROPOSED SYSTEM

The proposed system architecture contains the three phases known as DEV, STG and PRD. These three phases are explained here with some of the tools required to work in these phases [14].

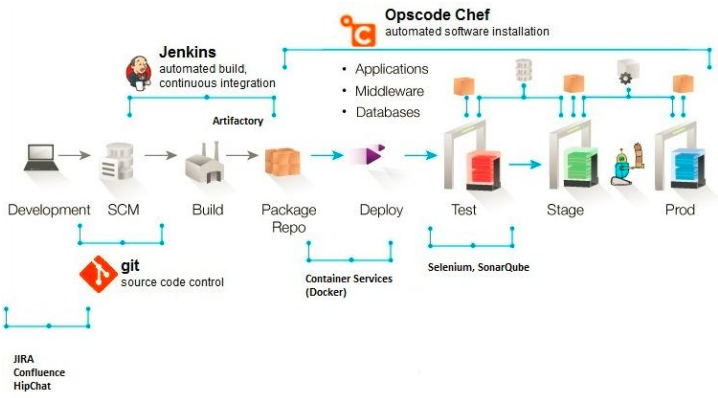

Figure 8: Architecture of Proposed DevOps System

A. DEV:

Dev is the kind of server in which the complete development of the project is done and uses various tools in this phase. After completion of the development process the project is sent for testing and this is tested in the test server. In some cases, Dev is also used for testing purposes like debugging [13].

B. TEST

Test is also a kind of server in which the testing takes place for the developed project. This testing will have some test cases written for testing of project. Testing can be manual or automated.

C. STG

STG also known as staging in which all the test cases passed and the project configuration and acceptance testing by the customer is done. If the customer agrees then it goes to deployment otherwise it will be changed according to the instructions given by the client.

D. PRD

PRD server is used for the production purpose. After accepting by the customer, the complete project after passing through all the tests will be deployed and released into the market.

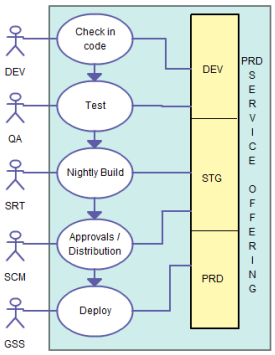

Figure 9: Dev, STG, PROD structure

TOOLS USED

1. APACHE MAVEN

It is a Software Management tool works with Project Object Model (POM). It is used as a reporting, building, documenting from an informational central source point and can be used for building Java based projects.

2. GIT SOFTWARE

Git is a version control system for tracking the changes in the local machine or the distributed system among the multiple people in the project. Its primary use is to Source code management (SCM) in software development but it is able to track the any number of changes done in the project. In distributed system its aim is to provide the speed, data integrity, availability and support for the distributed and non-linear work flows.

3. JENKINS

Jenkins is a continuous integration (CI) server or a tool written in java language. It is open source software for download. The continuous integration services can be provided for software development which can be done via command line or web application server.

4. ANSIBLE

Ansible is the automation software that provides configuration management, software provisioning and application deployment. A platform was created by the Michael Dehaan, who is the author of the provisioning server application cobbler and a co-author of the funcframework for remote administration. It is the part of Fedora Linux which is owned by Red Hat Inc., and is available for the Red hat Enterprise Linux, Cent OS, Oracle Linux via extra packages for Enterprise Linux (EPEL), Scientific Linux and for remaining OS's also. It is used for automating the tasks like software installation etc.,

VI. RESULTS AND DISCUSSION

The results for the Maven, Git, Jenkins and Ansible are shown here with the detailed steps for implementation of the system.

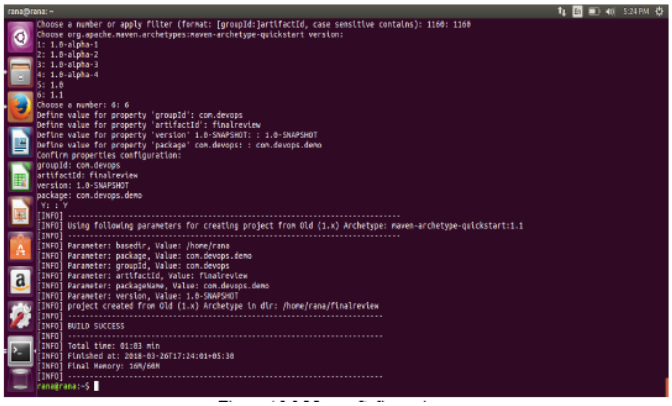

Figure 10: Maven Configuration

For configuring the Maven, we need to follow these steps. First open the Linux terminal and install the maven by typing "**sudo apt-get install maven**" in terminal then Create project hierarchy in maven by typing "**mvn_archetype:generate**".

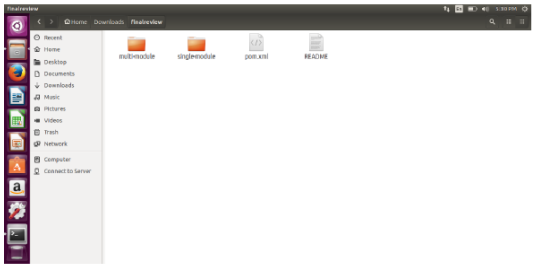

Figure 11: Local folder structure

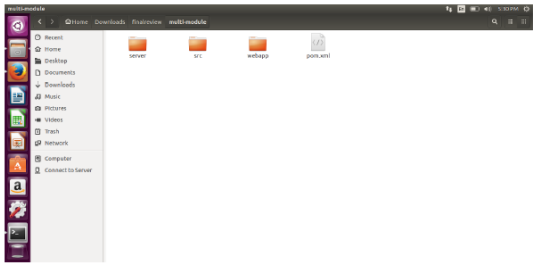

Figure 12: Multi module structure folder

After this it will prompt a window to create and configure project. The local folder structure of the project is shown in the figures 11, 12, 13. After that write the code in src directory of the project and update xml file.

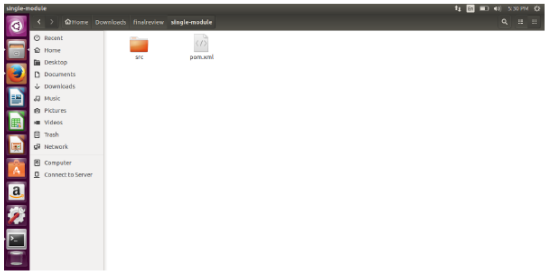

Figure 13: Single project folder

Git Installation is done by the following process. Firstly, open the terminal and type the following command "**sudo apt-get install git**"

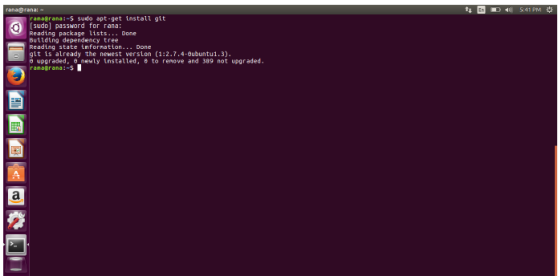

Figure 14: Git Installation

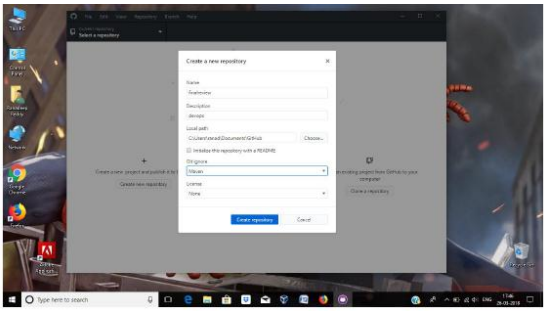

Figure 15: Creating new repository in GitHub Desktop

Download and Install GitHub Desktop for windows to pushing complete system from local repository to GitHub repository.

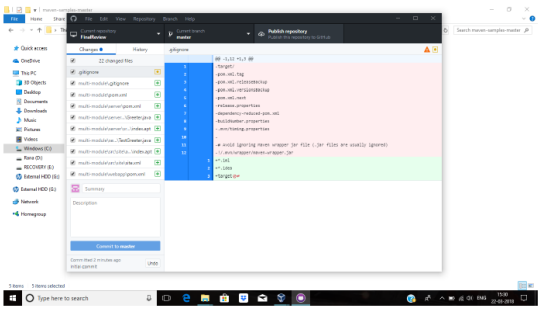

Figure 16: Uploading the files to GitHub repository

After creating the new repository click on open repository and copy the files from project workspace to GitHub Desktop workspace.

F

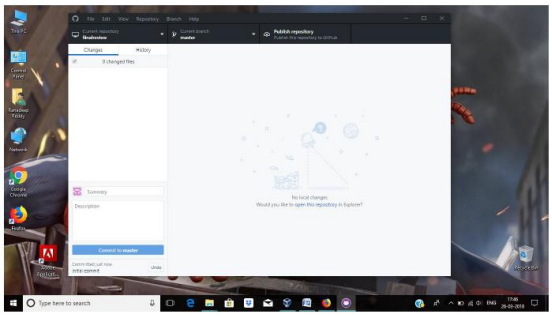

Figure 17: Publishing and uploading project

After copying the files from local repository to GitHub repository publish the repository and upload the project.

Installation of Jenkins is required for the Continuous Integration. To install Jenkins, we need to follow these steps. Firstly, open the Linux terminal and type the command "**sudo apt-get install jenkins"** to install Jenkins.

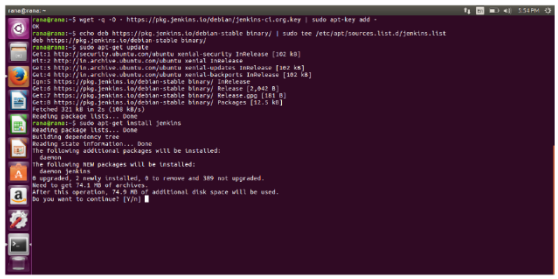

Figure 18: Jenkins Installation

After installing Jenkins open the Mozilla Firefox browser and type this in URL bar **"localhost:8080"** then create an account and login into Jenkins

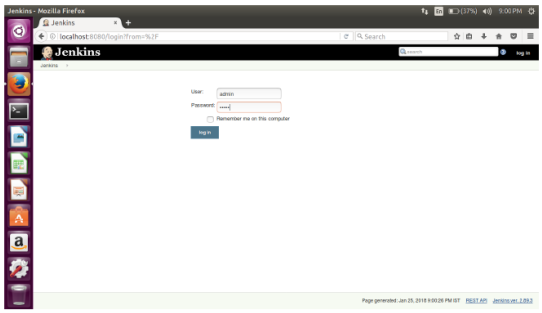

Figure 19: Login Page of Jenkins UI

After successful login into Jenkins, Click on Manage Jenkins link and go to home page of Jenkins to create a job.

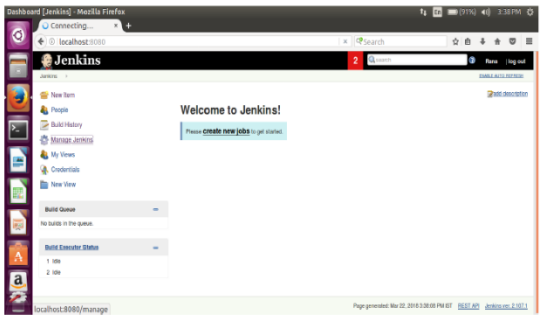

Figure 20: Jenkins Homepage

In Jenkins homepage, select the Maven project and enter the name of the project.

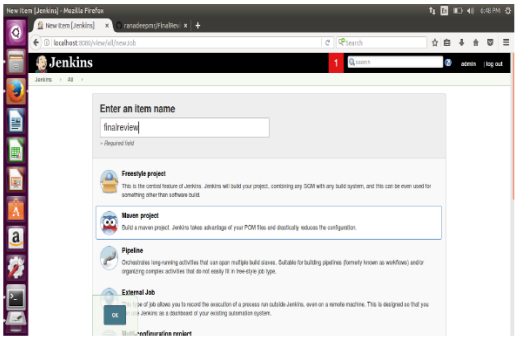

Figure 21: Creating new project in Jenkins

Copy the link from the GitHub repository. Select the Clone or Download button from the GitHub repository page.

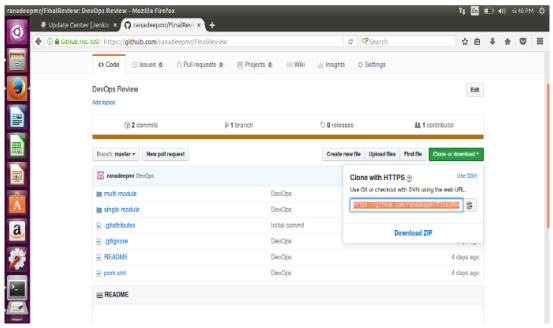

Figure 22: Copying / Cloning the GitHub repository

Paste the link copied from the GitHub and paste it in the Source Code Management as Git

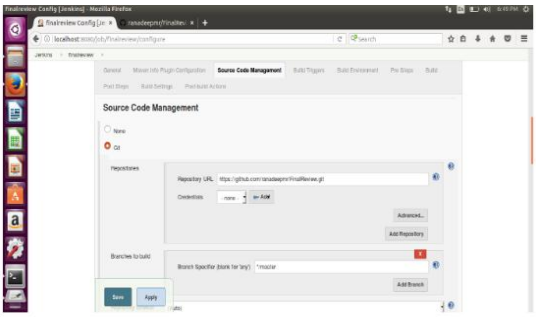

Figure 23: Source Code Management

After the source code Management, the project is ready to deploy. The Ansible is the tool used for configuration management, software provisioning etc., Here is the procedure of installation of ansible and making the process automate.

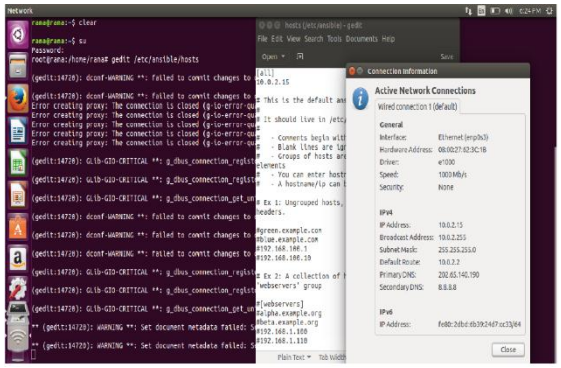

Figure 24: Finding IP Address

Firstly, open the Linux terminal and type the "**sudo apt-get install ansible**" command then Login as a super user then Type "**gedit /etc/ansible/hosts**" command and set the IP Address of your system. Create a .yml file which consists of list of commands. To automate ansible tool with single command **"gedit mymodule.yml".**

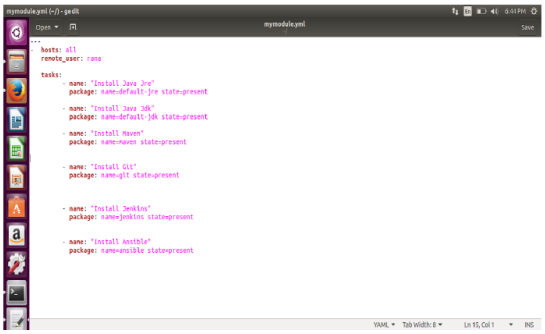

Figure 25: Automated files

After creating mymodule.yml file run the following command to automate tasks:
"**sudo ansible-playbook mymodule.yml**"

The results of this paper are shown above is the process of implementation of DevOps architecture by using tools like Maven which reports, documents the data in the project, Git used for version controlling and committing changes by pull and push requests into the Git. Jenkins is used for the Continuous integration and Ansible is the tool which is used for configuration management. The project will be first developed in the DEV server, testing and changes are done in the STG server and the deployment of the project is done in the PROD server.

VII. CONCLUSION and Future Scope

This proposed work is successfully designed, implemented and tested. DevOps (a portmanteau of development and operations) is a software development methodology that escalates to the amalgamation between software developers and information technology (IT) operation professionals. Its focuses mainly on delivering software product faster and reducing the failure rate of releases to make the product efficient. This system will be helpful for the developers or testers who need to fix the bugs rapidly and want to add extra features to the existing product according to the client requirement. At present DevOps is the most advanced approach in IT industry than waterfall model and agile model. This system can be extended by on boarding the complete project into cloud services like Microsoft azure or Amazon Web Services (AWS) etc. to improve the efficiency. This can also be extended by generating the review reports of the project through Data visualization tools like Power BI or Tableau for better understanding of the project to the client.

ACKNOWLEDGMENT

We would like to acknowledge the contribution of all the people who have helped in reviewing this paper. I Chellamalla Mamatha would like to give my sincere thanks to S C V S L S RAVI KIRAN the co-author for his support throughout the project and in writing this paper. We would also thank our families and friends who have supported us while writing this paper.

REFERENCES

[1]Carmine Giardino, Nicolò Paternoster, Michael Unterkalmsteiner, Tony Gorschek and PekkaAbrahamsson, "Sofware Development in Startup Companies: The Greenfield Startup Model", IEEE Transactions on Software Engineering, 2016

[2]Youssef Bassil, "A simulation Model for the Waterfall Software Development Life Cycle", International Journal of Engineering & Technology (iJET), ISSN: 2049-3444, Vol. 2, No. 5, 2012

[3]Dimitris Karagiannis, "Agile Modeling Method Engineering",Faculty of Computer Science, University of Vienna.

[4]David P. Harvie, Arvin Agah, "Targeted Scrum: Appling mission command to Agile Software Development", IEEE Transactions on Software Engineering, 2016.

[5]Dan Hao, Lu Zhang, Lei Zang, Yanbo Wang, Xingxia Wu, Tao Xie, "To be optimal or not in Test-case prioritization", IEEE Transactions on Software Engineering, 2016.

[6]Earl T. Barr, Mark Harman, Phil McMinn, MuzammilShahbaz, Shin Yoo, "The Oracle Problem in Software Testing: A Survey", IEEE Transactions on Software Engineering, 2015.

[7]Lucy Ellen Lwakatare, PasiKuvaja and Markku Oivo, "Dimensions of Devops", Springer International Publishing Switzerland 2015.

[8]Manish Virmani, "Understanding DevOps & bridging the gap from continuous integration to continuous delivery", Innovative Computing Technology (INTECH), Fifth International IEEE Conference, 2015.

[9]S. R. Chidamber ; C. F. Kemerer, "A metric suite for Object Oriented Design", IEEE Transactions on Software Engineering, 2015.

[10]Maximilien de Bayser, Leonardo G. Azevedo, Renato Cerqueira, "The case for Devops in scientific applications", IFIP/IEEE International Symposium on Integrated Network Management (IM), 2015.

[11]Michael Huttermann, "DevOps for Developers", Springer International Publishing Switzerland, vol. 1, 2012.
[12]Sungjoo Kang, Okjoo Choi and JongmoonBaik, "Model-based Dynamic Cost Estimation and Tracking Method for Agile Software Development, IEEE/ACIS Transactions on Software Engineering, 2010.
[13]M. WasifNisar, Yong-Ji WANG, ManzoorElahi, "Software Development Effort Estimation Using Fuzzy Login - A Survey", IEEE Transactions on Software Engineering, 2008
[14]L. Dobrica, E. Niemela, "A survey on software architecture analysis methods", IEEE Transactions on Software Engineering, 2002.
[15]A. Remi Jullian, M.Sangeetha, "From Dev to Ops – Introduction to Devops on understanding Continuous Integration and Continuous Delivery", International Journal of Innovative Research in Computer and Communication Engineering Vol. 4, Issue 6, June 2016.

Authors Profile

Ms. Chellamalla Mamatha working is an assistant professor in CVR College of Engineering with 1 year experience in teaching. Completed her Master's degree from CVR College of Engineering, an autonomous institution and JNTUH affiliated, NBA and NAAC accredited institution with the specialization of cryptography. Has worked for projects in the area of Security in Electronics Corporation of India Limited (ECIL). She completed her Bachelors from the Avanthi Institute of Engineering and Technology affiliated to JNTU Hyderabad..

Mr S C V S L S RAVI KIRAN is working as a Software Engineer in MAQ Software. Completed his Bachelors in CVR College of Engineering, an autonomous institution and JNTUH affiliated, NBA and NAAC accredited institution. Published many articles in an International and UGC approved journals. Done many projects on Internet of Things (IoT) and the work were published in international journals.

