

## Enhancement of CI/CD Pipelines with Jenkins BlueOcean

S.F. Rayanagoudar<sup>1\*</sup>, P. S. Hampannavar<sup>2</sup>, J.D. Pujari<sup>3</sup>, V. K. Parvati<sup>4</sup>

<sup>1,2,3,4</sup> Dept. of Information Science and Engineering, SDM College of Engineering and Technology (Affiliated to VTU, Belagavi), Dharwad, Karnataka, India.

\*Corresponding Author: [sfrayanagoudar@gmail.com](mailto:sfrayanagoudar@gmail.com)

Available online at: [www.ijcseonline.org](http://www.ijcseonline.org)

Accepted: 03/Jun/2018, Published: 30/Jun/2018

**Abstract** - This paper illustrates how BlueOcean embraces the Jenkins CI/CD pipelines for providing enhanced and better user experience by enabling teams to easily adopt to continuous delivery. Blue Ocean plugin brings out an enhanced user interface to Jenkins automation tool based on modern and personalized design that allows users to graphically generate, diagnose and visualize continuous delivery pipelines. With the introduction of visual pipeline editor not just the developer but any member of the DevOps team can easily understand the process flow, enabling them to quickly identify and explore the issue that's causing hindrance in execution of the flow.

**Keywords:** Continuous Integration, Continuous Delivery, Jenkins, Pipeline, BlueOcean, Jenkins File.

### I. INTRODUCTION

#### A. Continuous Integration (CI)

Continuous integration (CI) is a software development practice that requires developers to integrate code into a shared mainline or repository at a regular intervals. The main objective of CI is resolving integration problems in a project [8]. The developers should have a frequent builds in continuous integration. The common practice is that whenever a code is committed to a repository, build should be triggered [1].

#### B. Continuous Delivery (CD)

Continuous delivery is a DevOps software development approach in which teams produce software in a short sprints, ensuring that the software can be reliably released in production [1]. With proper implementation of continuous delivery developers will have deployable build artifact that has passed a standardized test process [3].

#### C. Continuous Deployment

Continuous deployment is a software development practice in which every code change is passed to the entire pipeline and is automatically put into production with many production deployments every day [2].

#### D. Jenkins

Jenkins is a powerful open source (OpSo) CI platform which supports building and testing of projects virtually. The Jenkins build system is written in java and provides 929 plugins for automation. Jenkins can be easily integrated with number of testing and deployment

technologies. The central build takes place on a server where Jenkins is installed. If the build fails or not working, Jenkins will send a notification to the developer. It might take 10min to 1 hour depending upon the complexity of the project. Thus a quick feedback or response is an advantage of Jenkins.

#### E. BlueOcean

Blue Ocean rethinks the user experience of Jenkins. Designed from the ground up for Jenkins Pipeline, but still compatible with freestyle jobs, Blue Ocean reduces clutter and increases clarity for every member of the team. Blue Ocean's main features include:

- **Sophisticated visualizations** of continuous delivery (CD) Pipelines, allowing for fast and intuitive comprehension of your Pipeline's status.
- **Pipeline editor** - makes creation of Pipelines approachable by guiding the user through an intuitive and visual process to create a Pipeline.
- **Personalization** to suit the role-based needs of each member of the team.
- **Pinpoint precision** when intervention is needed and/or issues arise. Blue Ocean shows where in the pipeline attention is needed, facilitating exception handling and increasing productivity.
- **Native integration for branch and pull requests**, enables maximum developer productivity when collaborating on code with others in GitHub and Bitbucket.

This paper is intended for beginners to get overall understanding of creating Jenkins pipeline project with enhanced user experience, modeling and presenting the process of software delivery by surfacing information with few clicks as possible used by development team with BlueOcean plugin.

The rest of the paper is organized as follows. Section II describes related work and motivation. Section III discusses the theoretical concepts of Jenkins workflow. Section IV introduces to the proposed methodology. Section V describes the implementation part. Section VI presents comprehensive experiment results. Section VII concludes the paper.

## II RELATED WORK

Valentina Armenise et al [1] illustrated how Jenkins is transformed from continuous integration platform to continuous delivery platform involving new feature where build, release and delivery of the product is automated. This paper not only describes Jenkins as a hub for continuous delivery but also introduce the challenges which need to be solved to strengthen Jenkins.

Mojtaba Shahin, Muhammad Ali Babar, Liming Zhu et al [2] describe about continuous practices i.e., continuous integration, delivery and deployment which are involved in software development practices. This paper involved systematic review on approaches, tools, challenges and practices used for future research. Authors have used SLR (systematic literature review) method for reviewing the peer-reviewed papers on continuous practices reviewing the peer-reviewed papers on continuous practices

K Sree Poornalinga, P Rajkumar [8] demonstrates the newly structured cloud infrastructure for complete automation of continuous integration, deployment and delivery. Authors have described the analysis to show proposed pipeline facilitating productive environment for the team to automate the build and deploy their code in production saving cost and time by increased software quality and productivity.

## III. THEORY THE JENKINS WORKFLOW

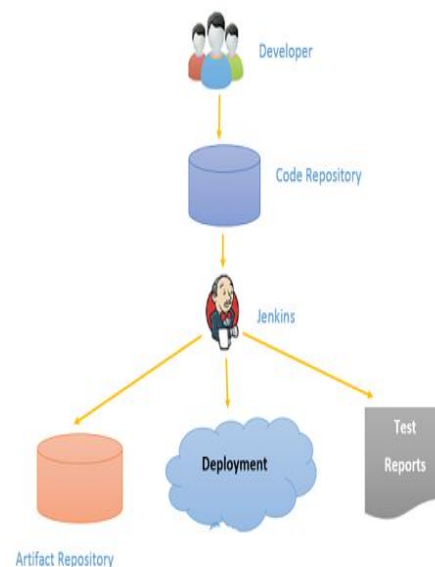


Fig.1 .Jenkins Workflow

Fig.1 describes how Jenkins is responsible for the CI/CD process. Initially developers will check their source code and commits the new modified code to the repository. Jenkins will pick up the changed source code from the repository and triggers a build. This build will run all the unit tests if required. The Jenkins dashboard contains the output of each and individual stable build after running successfully. The developer also gets the automatic notification if the build fails or something is not working properly in the build process.

### A. Jenkins Pipeline

A Jenkins pipeline supports you to describe an entire project life cycle as code. A CD (continuous delivery) pipeline is an automated statement of your mechanism for getting software from version control right through to your customers and users. Each and every modification to your software (committed in version control) goes through a complex mechanism on its way to being released [4]. This mechanism involves building the software in a repeatable and reliable way, as well as progressing the complete built software (also known as "build") through collective stages of testing and deployment. The description of a Jenkins Pipeline is documented into a text file (known as Jenkinsfile) which later can be committed to a project's version control repository [6].

### B. JenkinsFile

Jenkinsfile is a text file that includes the description of a Jenkins Pipeline and is checked into version control.

#### IV. PROPOSED METHODOLOGY

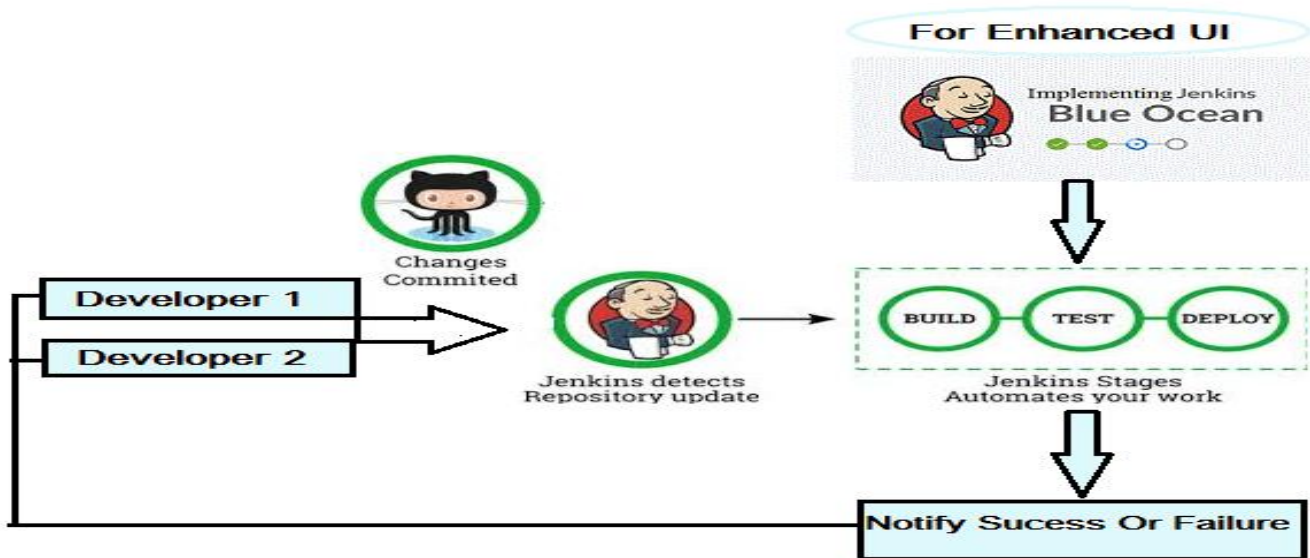


Fig.2 Flow diagram for proposed methodology

Fig.2 represents the proposed methodology for implementing the BlueOcean plugin for CI/CD pipelines. Jenkins community and Cloudbees have associated to architect and develop the Blue Ocean user interface with key features such as: Pipeline Visualization, Visual Pipeline Editing, Pinpoint Troubleshooting, GitHub and Git Integration, Built with and for Declarative Pipeline. The proposed methodology helps implement all the above said features with the help of Jenkins BlueOceanplugin

#### V. IMPLEMENTATION

##### A. Installation of Jenkins:

The WAR (Web application Archive) file version of Jenkins can be installed on any platform or operating system that is compatible with java.

To download , install and run the WAR (Web application Archive) file version of Jenkins tool:

1. The latest and stable Jenkins WAR file should be downloaded from the url: <https://updates.jenkins-ci.org/download/war/> to an appropriate directory on your machine.
2. Open up a command prompt/terminal window to the installation directory.
3. Run/Execute the following command :  
java -jar jenkins.war.
4. Browse to the following url <http://localhost:8080> and stay until the Unlock Jenkins page is loaded.
5. Blue Ocean plugin can be installed separately via Manage Plugins tab in Jenkins tool.

##### Note:-

The Jenkins port can be configured with different number by specifying the httpPort field when you execute/run the java -jar Jenkins.war command. For example to access Jenkins from port 9090, run the following command:

```
java -jar Jenkins.war -httpPort=9090
```

##### 1) Installing plugins in Jenkins

When the Jenkins is fully up and running, we need to implement concept of pipelines for projects by installing the plugins. This can be done via:

Manage Jenkins > Manage plugins > Available> Install all essential plugins. Ex: Git, GitHub, Pipeline, BlueOcean >Restart Jenkins after installation of plugin.

The new Jenkins BlueOcean and Pipeline project can be created once the Jenkins setup is ready.

##### B. Creating Project with Jenkins Blueocean.

To create a pipeline project with BlueOcean, GitHub Account with the respective project repository is required. This assures jenkinsfile is created with the support of BlueOcean visual editor available for the respective project or application.

### 1) *Creating a repository in GitHub.*

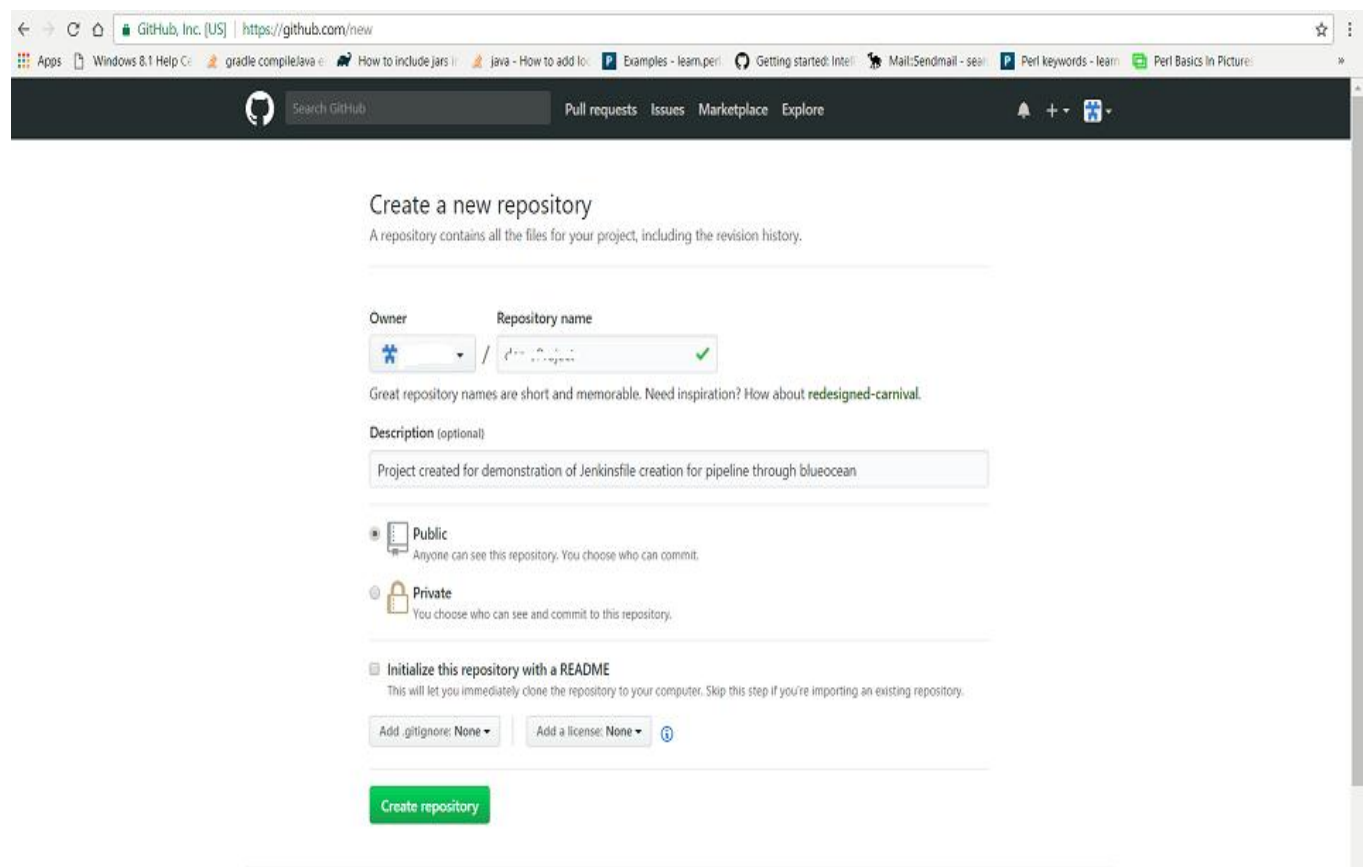


Fig.3 Creation of new repository in GitHub

Fig.3 represents the steps to create new repository in GitHub. A JenkinsFile is generated with the described pipeline syntax created through BlueOcean plugin for the respective project in the version control/repository. Therefore it is important to create a repository for project in GitHub as shown in the figure above.

### 2) *Creating Jenkinsfile with BlueOcean*

A JenkinsFile is created through the Jenkins BlueOcean for the respective project in the Repository. BlueOcean plugin ensures better user experience through its visual editor for the users. Each stage of BlueOcean can be easily defined through the user interface (UI) and corresponding

steps can also be added. The process of creation of JenkinsFile is run and saved through BlueOcean visual editor once all the stages are defined. This process completes the process of generation of JenkinsFile through the BlueOcean Plugin. The project starts to build the artifact and the results are displayed as shown in the figure 4. Of Experimental Results.

If the execution flow contains any bug /error, the process terminates the execution at a specific stage and user gets the notification regarding the failure. The user can view the error logs of only that particular stage to handle the issue.

## VI. EXPERIMENTAL RESULTS

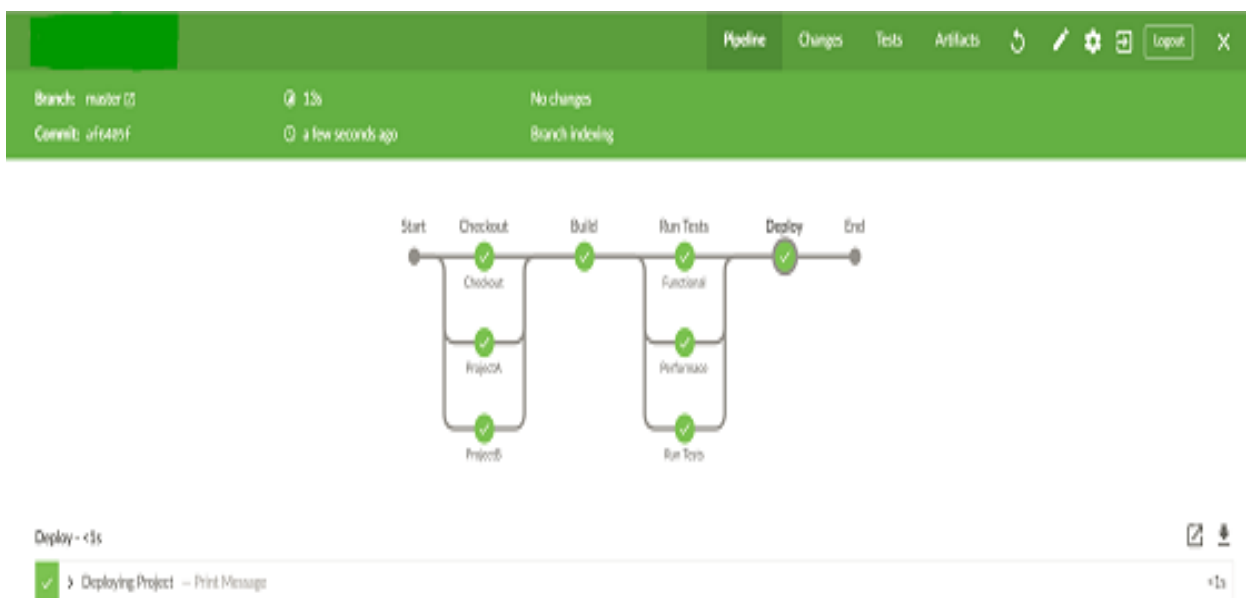


Fig 4. BlueOcean Pipeline showing the execution of project

Fig 4. Represents the experimental results through the BlueOcean Pipeline. The Jenkins pipeline precisely describes all the stages and the associated task with each stage. Separate logs for each stage is provided. If any error/bug occurs at a specific stage the user is notified with the error stage. The user can view error logs of only that particular stage which will ease the process of debugging.

## VII. CONCLUSION

Blue Ocean rethinks the user experience of Jenkins. It simplifies usage by making the UI more concise. Provides improved pipeline visualizations, providing clarity by taking user to the source of where a build fails in the code and also facilitates automated discovery of branches and validation of pull requests upon dropping a Jenkinsfile into the Git repository easily through the enhanced user interface by the plugin. Thus, it offers a personalized dashboard for projects that user choose to follow, making it easier to see at a glance what jobs need attention making the process of execution more efficient.

## ACKNOWLEDGEMENT

We would like to thank every member of the faculty at SDMCET for their guidance and support, which has helped us, complete this technical paper successfully.

## REFERENCE

- [1] Valentina Armenise, "Continuous Delivery with Jenkins: Jenkins Solutions to Implement Continuous Delivery", Release Engineering (RELENG), 2015 IEEE/ACM 3rd International Workshop, 19 May 2015, Date Added to IEEE Xplore: 30 July 2015.
- [2] M. Shahin, M. Ali Babar, and L. Zhu, "Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices", IEEE Access, 2017.
- [3] P. M. Duvali, S. Matyas and A. Glover, "Continuous Integration Improving Software Quality and Reducing Risk", Addison Wesley Signature Series publisher, UK, pp. 12-20, 2007
- [4] J. McAllister, "Mastering Jenkins", Packt publication, Sweden, pp. 58-70, 2015.
- [5] Mathias Meyer, "Continuous Integration and Its Tools", IEEE Software, 2014.
- [6] M. Soni, "Jenkins Essentials", Packt publication, India, pp. 30-45, 2015.
- [7] Manish Virmani, "Understanding DevOps & Bridging The Gap From Continuous Integration To Continuous Delivery," 2015.
- [8] K. Sree Poornalinga, P. Rajkumar, "Survey on Continuous Integration, Deployment and Delivery in Agile and DevOps Practices" International Journal of Computer Sciences and Engineering, Volume: 4, Issue: 4 PP(213-216) April 2016, E-ISSN: 2347-2693

### Authors Profile

**Ms. Shilpa F Rayanagoudar** obtained her Bachelor of Engineering degree from SDM College of Engineering and Technology (Affiliated to VTU, Belagavi), Dharwad, Karnataka, India in 2015 and currently pursuing Master of Technology in Information Technology from SDM College of Engineering and Technology (Affiliated to VTU, Belagavi), Dharwad, Karnataka, India. Her passion is in the field of DevOps, Agile Methodologies and data analytics.



**Ms. Priya S Hampannavar** is currently pursuing Master of Technology in the field of information technology from SDM college of Engineering and Technology (Affiliated to VTU, Belagavi), Dharwad, Karnataka, India. She obtained Bachelor of Engineering degree from SDM College of Engineering and Technology (Affiliated to VTU, Belagavi), Dharwad, Karnataka, India in 2015.



**Dr. Jagadeesh D. Pujari** is a Professor and Head in the department of Information Science and Engineering at SDM College of Engineering and Technology, Dharwad. He obtained his Bachelor of Engineering in Computer Science and Engineering from SDM College of Engineering and Technology, Dharwad, Karnataka University, Karnataka, India. He received his Master degree in Science from BITS, PILANI, India. He received his Ph.D. from Gulbarga University, Gulbarga, Karnataka, India. He has published 55 papers at International Journal and conferences and 03 at National conferences. He has conducted 06 workshops and 01 Conference.



**Prof Vasudev K. Parvati** is an Associate Professor in the department of Information Science Engineering, at S.D.M. College of Engineering and Technology, Dharwad, Karnataka, INDIA. He obtained his Bachelor of Engineering from SDM College of Engineering and Technology, Dharwad, Karnataka University, Dharwad, Karnataka, INDIA. He received his Master's degree in Technology from VTU, Belagavi, Karnataka, INDIA. He is pursuing Ph.D. from Karnataka University, Dharwad, Karnataka, INDIA. He has published 05 papers at National and 06 in international journals.

