IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-7, Issue-3, March 2019 E-ISSN: 2347-2693

Standard DevOps Pipeline

Mahadevi S. Namose[1*], Shobha D. Patil[2]

[1,2]Computer Engineering Department, JSPM NTC Narhe Pune-41, Pune University, Pune, India

[*]*Corresponding Author: mahadevi.sggs@gmail.com, Tel.: +919657853085*

DOI: https://doi.org/10.26438/ijcse/v7i3.641646 | Available online at: www.ijcseonline.org

Accepted: 14/Mar/2019, Published: 31/Mar/2019

***Abstract*—** The main focus was to enhance the potency, quality, and speed to promote at intervals the software system development world. First it was Waterfall, next it absolutely was Agile and currently it's DevOps. This is how today developers approach towards building the great products. DevOps is a way to develop software which consists of continuous development, continuous testing, continuous integration, continuous deployment, and continuous monitoring of the software throughout its development lifecycle. This is the way of process adopted by all the top companies to develop high-quality software and shorter development lifecycles. The goal of Continuous Delivery is to modify a relentless flow of changes into production via an automatic computer code line. The pipeline ought to give feedback to the team and visibility into the flow of changes to everybody concerned in delivering the new features. The delivery pipeline is countermined into some stages, as mentionedbelow.1.SourceCodeManagement 2.Continuous Integration 3.Static Code Analysis 4.Build the Artifact 5. Artifact Deployment.

***Keywords*—** Continous integration, Build, Artifacts, Log Monitoring, Test.

I. INTRODUCTION

DevOps is defined as a combination of software development and operations — and as its name suggest, it is a melding of these two disciplines in order to emphasize communication and collaboration between the traditionally separate developer and IT operations teams. Rather than seeing these as two distinct groups who are responsible for their specific tasks but don't really work together, the DevOps methodology recognizes the interdependence of the two groups. By integrating these functions as one team or department, DevOps helps an organization deploy software more frequently, while maintaining service stability and gaining the speed necessary for more innovation. And, in the end, everyone is able to deliver the best results and overall experience possible to the customer.Extending the Agile movement. The specific goals of DevOps span the entire delivery pipeline. They include improved deployment frequency, which can lead to: Faster time to market; Lower failure rate of new releases; Shortened lead time between fixes; Faster mean time to recovery (in the event of a new release crashing or otherwise disabling the current system) In simple words, DevOps can be defined as an alignment of development and IT operations with better communication and collaboration.

In this paper authors give the brief introduction of Devops in the first part, In the second part we will discuss Devops tools challenges and benefits. In the third part we will focus on the system design (Delivery Pipeline Steps).In last part we will see Discussion and conclusion.

II. CHALLENGES AND BENEFITS

1. *Challenges in Standardization of DevOps*

1. *Tools Challenges*

- Multiple tools are used to evaluate with one or more overlapping features in the tools, learning curve involved to evaluate/do proof of concept and choosing the tools with short-term goals instead of long-term perspective.
- Limited knowledge of tools, used from the perspective of automation in their specific area
- Lack of out-of-box integration with most tools used in the CD pipeline and teams investing time to build custom integrations.

2. *Process Challenges*

- There is no central DevOps team who can evaluate and suggested the right set of tools to be used, with individual project teams.
- There is no defined process framework detailing the different tenets in DevOps.
- There is a lack of knowledge on defining right KPI for each DevOps capability and how to measure the metrics to determine efficiency of the DevOps

automation process and identify scope of optimization.

- There is a lack of SME knowledge on the application.
- Multiple vendors are managing the different stages of an application and each following their own defined process for development life cycle.

3. *Cultural Challenges*

- Standardization on metrics, process, practices, governance and organization-level control over DevOps implementation is challenging without a central driving these practices across the enterprise.
- Teams operate in silos and follow their own DevOps practices and think in silos without having a holistic view, resulting in broken processes that are inconsistent.
- Teams still lack training in DevOps on the tools, practices to be followed in CI, continuous testing, continuous deployment and operations.
- Lack of DevOps skilled resource brings challenges in implementing end-to-end automation
- There is no standard approach for adoption of Devops andwhere to start and how to start because DevOps is considered a concept.

2. *The benefits of DevOps*

The main benefit of Devops is- with a single team composed of cross-functional members all working in collaboration, DevOps organizations can deliver with maximum speed, functionality, and innovation. Devops benefits are classified into three category-

1)Technical Benefits

2) Cultural benefits:

3) Business Benefits

*Technical benefits***:**

- Continuous software deliveryShorter release cycles lead to shorter feedback cycles. Your customers are receiving frequent and timely updates, which gives you the ability to evaluate and test their satisfaction and response early on.
- Less complexity to manage
- Efficient operations ,Agile and DevOps are built on lean principles. The reduction of waste, such as unnecessary waiting times or process overhead through continuous improvement, is a core principle.This not only helps agility, but also has a positive impact on costs by helps to limit their growth.
- Increased service quality
- Increased reliability of service delivery The perceived timeliness of delivery is an extremely important satisfaction factor for your internal or external customers Usable results are delivered early and frequently, which is critical in uncertain and changing business environments.

4. *Cultural benefits:*

- Happier, more productive teams
- Better usability increases customer satisfaction
- You can even test different versions of your product at the same time with different customer groups by A/B testing. This gives you the ability to compare different new features or capabilities of your products to each other
- Higher employee engagement
- Greater professional development opportunities

5. *Business benefits:*

- Faster delivery of features
- More stable operating environments
- Improved communication and collaboration
- More time to innovate (rather than fix/maintain)
- Improved customer experience and satisfaction
- Enabling the business
- Digital transformation

3. *DevOps Principles*

There are six principlesof Devops:

- Customer-Centric Action: DevOps team must take customer-centric action for that they should constantly invest in products and services.
- End-To-End Responsibility: The DevOps team need to provide performance support until they become end-of-life. This enhances the level of responsibility and the quality of the products engineered.
- Continuous Improvement: DevOps culture focuses on continuous improvement to minimize waste. It continuously speeds up the improvement of product or services offered.
- Automate everything: Automation is a vital principle of DevOps process. This is not only for the software development but also for the entire infrastructure landscape.

- Work as one team: In the DevOps culture role of the designer, developer, and tester are already defined. All they needed to do is work as one team with complete collaboration.
- Monitor and test everything: It is very important for DevOps team to have a robust monitoring and testing procedures.

III. Delivery pipeline steps

Choosing the Right Set of Tools and Stack Tools selection should be based on the requirements of the project and skills of the employee. Figure 1 shows the Devops delivery pipeline steps.

1. Step 1: Coding

Developer can develop source code in their integrateddevelopment environment or text editor according to theirpreferences. Below are the some text Editors:

Best Collaborative Text Editors

- CodeShare
- Codeanywhere
- CodePen

Best Free Text Editors in 2018

- Atom
- Visual Studio Code
- Brackets
- Notepad++
- TextMate
- Vim
- Komodo Edit
- Eclipse
- InteliJI

Best Paid Text Editors

- Sublime Text
- CoffeeCup HTML
- Coda
- BBEdit

2. Step 2: Source Code Management

After coding, developer commit their changes viaGit, and then push them into GitHub.The reasonto choose Gitand GitHubis that Gitis an open sourcesystem as well as one of the most popular version controltools.GitHub is a web-based version-control and collaboration platform for software developers. Microsoft, the biggest single contributor to GitHub, Moreover, GitHuboffers free repositories on the clouds,therefore, developer do not need to setup the repositories locally. Git is used to store the source code for a project and it also stores the history of all changes done for that code. Multiple developer can manage the code at a time Public repositories are free for developer to change the code,private repository is not free, and it offers various paid plans.Each public or private repository contains all of a project's files, as well as each file's revision history.

3. Step 3: Continuous Integration

Once the source code is updated, Jenkins automaticallystarts the pipeline by the specific trigger or timer definedby developer or coder This step utilize the build and Test tools.

1)Jenkins:

Jenkins is one open source tool to perform continuous integration that monitors executions of repeated jobs, such as building a software project. if one of the steps fails Jenkins stop the execution. Jenkins also notifies the users about the build success or failure. Jenkins is java based ,simple and easy to install . Jenkins provides an impressive browser-hosted project management dashboard. Jenkins can be startedby two ways first is via the command line and second can run in a web application server. Under Linux you can also install Jenkins as a system service. It supports file fingerprinting, it also support security performance.

Through plugins Jenkins offers an extension of features .Its plugin list is very broad, and one can easily add their own plugins to that list. Jenkins can distribute software builds and test loads on several machines.

4. Step 4: Static Code Analysis

Sonar Qube performs a series of tests and output the analysis results.The Sonar platform analyzes source code from different aspects and hence it drills down to your code layer by layer, moving from the module level down to the class level. At each level, SonarQube produces metric values and statistics, which shows the problematic areas in the source that require improvement.

SonarQube mainly performs three kinds of tests.

- Static Test: SonarQube utilizes several static test tools to analyze the source code and discover the code smells, bugs and vulnerabilities. In addition, there are other languages supported apart from Java in SonarQube. Teachers can choose various kinds of plugins needed in their courses from the SonarQube plugin library.
- Unit Test: Students ought to write the test code and SonarQube will execute the test cases, finally, generates a report containing the results of tests.
- Coverage Test: Plugins such as JaCoCo11 and Cobertura12 are able to generate the code coverage reports which can assess whether the test cases are

sufficient. All of the test reports are synthesized and our system selects the necessary data which help the improvement of the code quality and the evaluation of the projects.

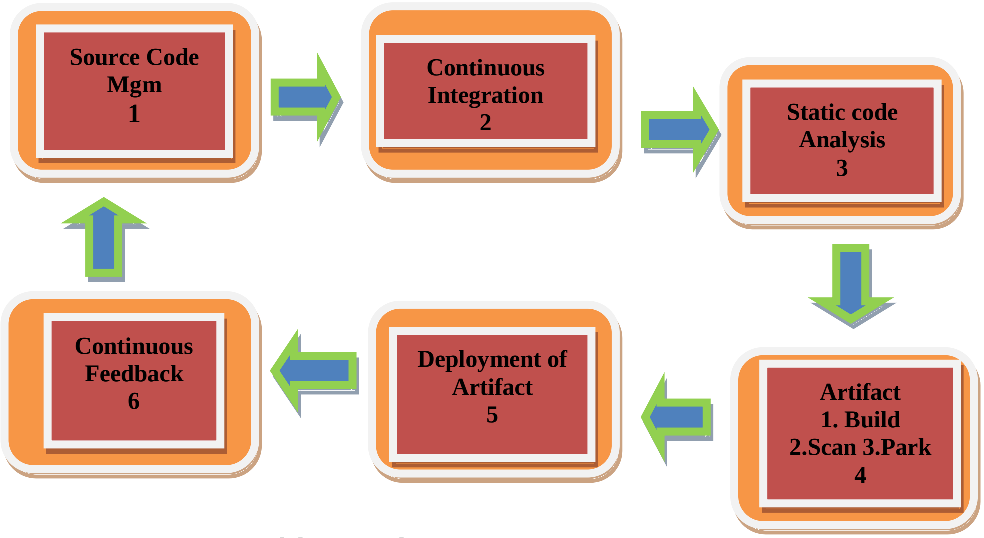

Figure 1.Devops delivery pieline

List of DevOps Tools for Testing

- MUnit: A Mule application testing framework that allows you to easily build automated tests for integrations and APIs. It provides a full suite of integration and unit test capabilities.
- SoapUI: An open source API testing tool for SOAP and REST APIs. Features include REST API functional testing, WSDL coverage, and functional testing for SOAP Web Services.
- JUnit: An open-source DevOps tool that is used widely. It is an automation server that is a part of Jenkins. This tool makes testing easier and more automated. Other tools include: Arma, Parasoft, Perfecto, and Zuul

5. *Step 5:Build The Artifact*

After testing, Jenkins builds the source code and generates artifacts via specific build tools.The stage after testing is to build. Jenkins will package the resource code via a specific build tool (e.g., Maven) and generatean artifact (e.g., WAR file). List of DevOps Tools for Building

- Gradle: An open-source automation software that can be used as a DevOps tool. Granadle helps users develop, test, deliver, and package applications onto any platform. The tool has a series of open source plugins and rich APIs.
- Maven: A popular, open-source tool for both development and testing. However it doesn't support Microsoft projects.
- MSBuild: A tool from Microsoft that allows users to easily compile and build projects and applications in a customized and automated fashion

6. *Step 6:Scan the Artifact*

As a part of standard DevOps pipeline we must scan the artifact to make sure it is not vulnerable. There are multiple tools available to scan the artifact. IQScan from SonaType is one of the popular tools to scan the artifact. IQScan does provide the scanned report in the form of document (PDF). Other tool can be used is Checkmarx, which is also getting popularity in the DevOps world. There is an easy way to integrate these tools with Jenkins.

7. *Step 7: Park the Artifact*

Once the artifact is generated successfully and scanned for security vulnerabilities with mentioned tool sets, the next thing to do is put the artifact somewhere for further usage or to deploy onto different environments e.g. DIT, SIT, UAT, PROD etc. Nexus would be the recommended and popular repository to keep the artifact and pull down whenever required.

8. *Step 8: Artifact Deployment*

Once we park the artifact onto Nexus, now we need to deploy it to the environment where we want to test it before moving it to production environment. There are many tools available to deploy the artifact e.g. Ansible, Chef, Puppet etc. We would recommend to useAnsible as an deployment tool.

List of DevOps Tools for Deploying

- *Puppet:* An open-source DevOps tool that provides users with foundations for DevOps practices, including automated testing, continuous integration, and continuous delivery.
- *Ansible/Chef:* A simple IT automation DevOps tool that allows users to automate solutions by making it easier to deploy systems and applications. The platform is comparable to Chef and Puppet.

9. Step 9: Continuous Feedback

Developer can modify their code based on thefeedback from user and start a new cycle from stepone.

IV.DISCUSSION

1. Customizability

Above system mainly integrates Jenkins, SonarQube and Docker along with version control systems and builds tools. We applied a hard-coded strategy to simplify development complexity and provide the system for students as fast as possible. This strategy inevitably impacts the customizability of system.

2. Metrics Definition

The set of metrics for evaluation is based on the data available in existing tools, which has not been defined with a broad consideration. With increased experience and understanding of DevOps education, we might need more metrics from various perspectives. In this sense, system needs to be enhanced to support defining more metrics. Besides, an automatic tool integrating different algorithms could be developed to generate the grades for students in diverse perspectives, for example, productivity, quality and contribution, etc[2].

3. Automatic Evaluation Tool

Most of the functions in above system are designed particularly for students to practice DevOps, such as team management and project management. Though the work status is analyzed and provided by our system, the evaluation is still a semi-automatic process rather than complete automation.

V.CONCLUSION

In this paper, we aimed to showcase, choosing right set of toolset is an integral part standard DevOps pipeline. Incorrect usage of tools can become an over headed DevOps pipeline.

In log monitoring tools combination of Logstash, Elastic search and Kibana is said to be the best tool. In system monitoring tools Graphite is said to be the best tool. In build and test tool Jenkins and Maven is said to be best tools. In Deployment and Configuration tool Ansible is considered to be the best tool.

REFERENCES

[1] Akshaya H L, NisargaJagadish S, Vidya J, Veena K, "*A Basic Introduction to DevOps Tools*", International Journal of Computer Science and Information Technologies, Vol. 6 (3) , 2015, 2349-2353 ISSN 0975-9646

[2] Guoping Ong, Shenghui Gu, He Zhang ,Dong Shao ,"*Devopsenvy: An Education Support System For Devops*",2017 ieee 30th conference on software engineering education and training (csee&t) 37-46

[3] Shwetha D, Swetha, Suhas K R, Devaraju J T, "*Relay Selection Algorithm for IPTV based Services",* IJSRNSC Vol.6 , Issue.1 , pp.1-6, Feb-2018

[4] Shruti Chopra,VijayKumar Joshi," *Gaming Prediction Analysis Using Hybrid Chi Square - Support Vector Machine Model and Historical Data*", Isroset-Journal (IJSRCSE) Vol.5 , Issue.5 , pp.1-5, Oct-2017

[5] P.V.Nikam,D.S.Deshpande,"*Different Approaches for Frequent Itemset Mining*", |Isroset-Journal (IJSRCSE)Vol.6 , Issue.2 , pp.10-14, Apr-2018

[6] T.SenthilSelvi,R.Parimala, *"Improving Clustering Accuracy using Feature Extraction Method"*,Isroset-Journal (IJSRCSE)Vol.6 , Issue.2 , pp.15-19, Apr-2018

[7] M. A. R, J. K. Patel, S. Akhtar, V. K. Agrawal, and K. N. B. S. Murthy,"*Docker container security via heuristics-based multilateral securityconceptual and pragmatic study",* in 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT '16). IEEE, 18 March 2016, pp. 1–14.

[8] A. A. Mohallel, J. M. Bass, and A. Dehghantaha, "*Experimenting with Docker: Linux container and BaseOS attack surfaces",* in 2016 International Conference on Information Society (i-Society '16). IEEE, 10 October 2016, pp. 17–21

[9] M. de Bayser, L. G. Azevedo, and R. Cerqueira,"*ResearchOps: The case for DevOps in scientific applications",* in 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM '15). IEEE, 11 May 2015, pp. 1398–1404

[10] D. P. Groth and E. L. Robertson, "*It's all about process: Project-oriented teaching of software engineering"*,in Proceedings of the 14th Conference on Software Engineering Education and Training (CSEET '01). IEEE, 19 February 2001, pp. 7–17.

[11] B. R. von Konsky and M. Robey, "*A case study: GQM and TSP in a software engineering capstone project"*,in Proceedings of the 18th Conference on Software Engineering Education Training (CSEET '05). IEEE, 18 April 2005, pp. 215–222.

[12] S. Jarzabek and P.-K. Eng, "*Teaching an advanced design, team-oriented software project course"*,in Proceedings of the 18th Conference on Software Engineering Education Training (CSEET '05). IEEE, 18 April 2005, pp. 223–230

[13] M. Raza, J. ao Pascoal Faria, and R. Salazar,*"Helping software engineering students analyzing their performance data: Tool support in an educational environment",* in Proceedings of the 39th International Conference on Software Engineering Companion (ICSE-C '17). Piscataway, NJ, USA: IEEE Press, 20 May 2017, pp. 241–243.

[14] G. Rong, H. Zhang, S. Qi, and D. Shao,"*Can software engineering students program defect-free?: An educational approach*", in Proceedings of the 38th International Conference on Software Engineering Companion (ICSE '16). New York, NY, USA: ACM, 14 May 2016, pp. 364–373.

[15] M. Ohtsuki, K. Ohta, and T. Kakeshita,"*Software engineer education support system ALECSS utilizing DevOps tools*", in Proceedings of the 18th International Conference on Information Integration and Webbased Applications and Services (iiWAS '16). New York, NY, USA: ACM, 28 November 2016, pp. 209–213.

[16] A. Crystal and B. Ellington,"*Task analysis and human-computer interaction: approaches, techniques, and levels of analysis*",In Proceedings of the Tenth Americas Conference on Information Systems, New York, Aug. 2004.

Authors Profile

Ms.M.S.Namose pursed Bachelor of Technology from University of **Swami Ramanand Teerth Marathwada University–** Nanded in 2009 and Master of Engineering from Pune University in 2014. She is currently working as Assistant Professor in Department of Computer Engineering in JSPM Narhe Technical Campus Narhe Pune-41,She has published 06 research papers in reputed international journals IEEE. Her main research work focuses on Automation Tool,Data Mining, IoT and Computational Intelligence based education. She has 6 years of teaching experience .

Ms S.D.Patil pursed Bachelor of Engineering from **Shivaji University in 2008** and Master of Engineering from Pune UniversityUSA in year 2015.She is currently working as Assistant Professor in Department of Computer Engineering in JSPM Narhe Technical Campus Narhe Pune-41,She has published 05 research papers in reputed international journals IEEE. Her main research work focuses on Automation Tool,Data Mining, Computational Intelligence based education. She has 5 years of teaching experience .