AUTOMATE INFRASTRUCTURE DELIVERY THROUGH DEVOPS

Dr.J.Dhilipan^a, D.B.Shanmugam^b, Dr.T.Prabhu^c, Dr. A.Raj kumar^d

^aProfessor & Head, Department of Computer Applications, SRM Institute of Science and Technology, Ramapuram Campus, Chennai.

^bAsst.Prof, Department of Computer Applications, SRM Institute of Science & Technology, Ramapuram Campus, Chennai.

^cAssociate Professor, Department of Computer Applications, Dr.MGR Educational & Research Institute, Chennai.

^dProfessor & Head, Department of Electrical and Electronics Engineering, K. Ramakrishnan college of Technology, Trichy.

Abstract

DevOps, instrument absorption, robotizes the assignments of programming advancement by overcoming any barrier between programming improvement and activities group. This exploration paper presents the need and phrasing of source code the executives instruments utilized during DevOps life cycle. Various apparatuses exist for every one of the product improvement stages which are then assessed dependent on choice rules and the device with the most elevated score is being picked. This paper audits elective source code the executives instruments viz. SVN, Git, Clear Case. This investigation will be valuable for specialists/understudies to comprehend the usual way of doing things of DevOps for its source code the executive's period of programming improvement.

Keywords: automation, automation tools, DevOps, software development, source code management

1. Introduction

A piece of software goes through various stages during its lifecycle of requirement analysis to production delivery. Most critical stages in the complete lifecycle is Development and Testing, where most of time is spent by different team members. Output generated by these two roles is – Code and Test cases/Test Scripts. Both (Code and automation script) of these are stored in multiple files across different logical directory structures. If there is only one person who needs to write the code/script, then there is no tool required for managing the code, this case is rare in the industry. While in each organization there are multiple teams and each team there are multiple members who need to work on software application code and need access to code at same time. Hence a tool is required to manage the code, control the access and maintain the versions of the code file across multiple directories. These tools are commonly referred as SCM (Source Code Management) Or VCS (Version Control Systems) tools. [17-19] Different SCM tools exist viz. SVN, GIT, Gerrit, GitLab, Bitbucket, Clear Case, GitHub etc. Work being reported herein involves the review of some of these tools that are employed by DevOps during the code management stage of the software development. This Paper also discusses the role of SCM tools in DevOps life cycle.

Literature Review

The practices or tools used by DevOps are reviewed separately from different existing research papers. One of the papers considers DevOps - a portmanteau of Development and Operations as a set of practices focused on using

latest generation tools to automate the configuration process for system resources and application components. This paper also accepts the improvement of process efficiency from hours and days to seconds and minutes. It also verifies that the longer an organization has implemented and continues to improve upon DevOps Practices, the better it performs. And better IT performance correlates to higher performance for the entire organization. [1]

Another paper verifies that the DevOps phenomenon emerged with the promise of easing the process of putting new software changes to production at a fast rate whilst also increasing the learning and innovation cycles of their products. They also focus on consolidating the understanding of DevOps and its practices as described by practitioners using multivocal literature and interviews. This study contributes to a scientific definition of DevOps and patterns of DevOps practices to help identify and adopt the phenomenon. [2]

One survey paper on DevOps Automation lists four research questions that motivated this work are which and how automated procedures support DevOps practices, which automation tools are being used to support DevOps teams, Which are the most important service characteristics that affect automatic service synthesis, What is the level of services synthesis automation? This series of research questions clarify many things regarding what DevOps actually is, how it can be automated and, finally, how it can be combined with one of the latest software development approaches, that is, software service composition. But this paper also admits less literature available on DevOps. [3]

This survey primarily relies on evaluating peer-reviewed papers, which is standard practice in the academic community for guaranteeing the quality and credibility of published works. Figure 1.1 depicts the evolution of published works on DevOps according to the criteria defined in our study design (papers collected in September 2018). Similar to most research areas, conferences generally publish preliminary results. In contrast, journals publish more mature studies. Academic journals primarily reach an academic audience, while magazines also have practitioners as their readers.



Fig. 1.1: Publications by source types and publication year

The "IEEE Software" magazine was a constant source of papers on DevOps, having published 13 of the 50 core papers. They are mostly relatively short papers describing real experiences faced by organizations. Therefore, it is advisable for practitioners to subscribe to this source for the latest updates on DevOps.

We did not find any peer-reviewed journal, magazine, or event dedicated exclusively to the theme. The closest related events we found were the "International Workshop on Release Engineering and the "International Workshop on Quality-Aware DevOps." For researchers beginning a study on DevOps, these are two potential venues for publishing their first results. The absence of a specific venue indicates an opportunity for the academic community to organize scientific events and journal special issues dedicated to this area.

2. SOURCE CODE MANAGEMENT TERMINOLOGY

Before proceeding further, first we'll review different terms used frequently during the coding phase of DevOps.

Repository

A code repository is a storage location used to keep code/configuration files of any software product. These files are organized in multiple directories in a logical manner and access can be controlled at directory or repository level. Single application code can be stored in a repository or across multiple repositories and each repository has its own unique URL.

When a developer makes a copy of code from a centralized SCM tool server and stores the same on his/her local machine in a similar structure, then this is called a local repository. Developers always make changes to local copy, make desired changes(Add/edit/delete) in code and then send the code to a centralized location that is called remote/Server copy and thereafter it becomes a single version of truth and can be distributed from there.[15-16]

Check-Out

This is a process of taking a copy of source code from centralized server to local machine for making changes. As checkout is always taken from the tool server, hence called as remote checkout as well.

Check-In

Developers make changes in local copy of code, test the logic and artifact build and then send the updated files to the centralized server so that the same can be accessed by other members of the team. This process is called Check- In.

Once a developer writes the code, it is advisable that someone review the code before it is checked-in to the repository. If developers are working in pairs where more than one developer writes the same piece of code after continuous discussion, then review is done along with coding, otherwise the developer creates a code review request. Some tools allow the review with SCM tool where review request before merge is called "Pull request", while in other case team use separate code review tool like Crucible.

Branch

When development of a software system starts, the first step in development is to create a repository in SCM, where developers check-in the code. This empty location is called the root of the repository and underneath there are trunk and branches. This terminology is taken from Tree, which starts from root and grows to trunk and branches. In SCM, the trunk is where the complete code of the system lies, in some tools it is also called as Master. [11]

Developer takes a copy of trunk/master code and creates a separate branch to both develop a new feature on the same code base and merge code back to truck. There are various such branching strategies that exist to manage the code effectively. In case, a branch is cut from trunk to release the code to production while development can continue in trunk. If there is any defect/incident reported in the production system then code fix can be done in a specific branch and release to production, while in some teams trunk based development model is

followed, which indicates there are no branches, and a single version of code lies in Trunk/Master.

Tags

Tags are a unique ID assigned to a group/batch of code. This is generally done to identify working copies of code that can be released to any environment like QA for testing or production. If there is any defect, then it is easy to identify a batch of files together and make changes in that batch rather than the latest copy of code which may have different code than a specific tag. Generally, tags are kept in sync with release versions.

3. ROLE OF SCM TOOLS IN DEVOPS

Key objective of DevOps is to enable continuous deployment by eliminating bottlenecks at every stage of the end to end development lifecycle. One of the key phases is Coding and Test Automation which generates code.

Managing the code generated by a team of developer working in parallel to delivery new features, fixing the defects received in non-prod environment, handing the production incidents and providing fix, tracking the versions of file, managing conflicts in code merge requests, handing different version of code: code in production, feature development code, code in QA testing. In short, a Development job is not just writing the code to satisfy functional and nonfunctional requirements, but also managing the code: versions, access, configuration etc. [14]

DevOps recommend automation at every stage; hence application code is not just functional code but build pipeline(Jenkins Build script) as code, Test Automation Scripts(Selenium code), Environment management scripts (Docker/Chef/Puppet/Ansible), Production monitoring scripts etc.[12]

So effective implementation of DevOps needs the most effective, efficient and optimized tool to manage the above mentioned type of code generated at every stage of the life cycle.

Following Diagram depict code generated at each stage of DevOps lifecycle, which needs to manage for effective implementation of DevOps. The figure 4.1 shows the source code depiction in DevOps.



Fig. 4.1: Source code depiction in DevOps

In Short, SCM tool is required to control, efficiently manage and version of code produced at every stage as shown above.

4. DISCUSSION AND CONCLUSION

As we have seen how different automation tools can manage the source code during DevOps lifecycle can accelerate the delivery process up to a great extent. Coding stage of the software development life cycle includes different code segments written by different developers. Hence it becomes utmost important to manage the complete code at a single place and make it available to all at the same time. Source Code Management tools are the best option for this task. There exist lots of management tools with their respective pros and cons. This research paper discusses the need and terminology for this management.

This paper also suggests the depiction of source code generated at each stage of DevOps life cycle. This depiction will not only manage the source code of the developed software efficiently but also improves the delivery time dramatically.

Also an analysis of the impact of using other tools may be carried out as a part of further research. This review will be used to design and follow the right set of automation tools to manage the source code during the software development according to particular system requirements.

References

- [1].Andreas Andreou, Andreas Christoforou, Luciano Baresi, Mike Papazoglou, Damian Tamburri, Deliverable D4.5 Survey paper / Technical report on DevOps Automation and Software Service Composition, "Dossier-cloud Devops-based Software Engineering For The Cloud"
- [2].Lucy Ellen Lwakatare, Pasi Kuvaja, Markku Oivo, "An Exploratory Study of DevOps Extending the Dimensions of DevOps with Practices", Faculty of Information and Electrical Engineering University of Oulu, Finland.
- [3]. https://au.pcmag.com/productivity/37713/atlassian-confluence
- [4]. https://jmeter.apache.org/
- [5].https://www.google.com/search?q=jira&source=Lnms&Tbm=Isch&Sa=X&Ved=0ahukewi-55_qkylhahvjfhikhap1cv4q_auidygc&Biw=1440&Bih=789#Imgrc=505rcq5w5mpfam:
- [6].http://www.softwaretestingtalk.com/selenium-webdriver/

[7].https://www.codeproject.com/Articles/1173711/Learn-about-code-review-in-Bitbucket-Cloud-2

[8].https://confluence.atlassian.com/bamboo/understanding-the-bamboo-ci-server-289277285.html

- [9]. https://www.atlassian.com/software/crucibleR.
- [10].Vaasanthi, V. Prasanna Kumari, S. Philip Kingston, "Analysis of Devops Tools using the Traditional Data Mining Techniques", International Journal of Computer Applications (0975 – 8887) Volume 161 – No 11, March 2017.
- [11].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.
- [12].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.