International Journal of Research in Engineering, Science and Management
Volume-2, Issue-5, May-2019
www.ijresm.com | ISSN (Online): 2581-5792

End to End Automation for Carrier Grade Telecom Products: A Case for DevOps

Siri Prakash[1], Minal Moharir[2]
[1]*Student, Department of Computer Science and Engineering, Rashtreeya Vidyalaya College of Engineering, Bangalore, India*
[2]*Assistant Professor, Department of Computer Science and Engineering, Rashtreeya Vidyalaya College of Engineering, Bangalore, India*

***Abstract*: DevOps, a combination of two words Development and Operations, is an extension of agile methodology. DevOps stresses the importance of communication, collaboration and integration between software developers and information technology professionals. The need to reduce the time between making changes to a system and delivering those changes to the production site is addressed by involving the concepts of Continuous Integration, Continuous Delivery, Continuous Deployment and Monitoring. We consider the case of telecom industry, which is increasingly transforming towards virtualized environments. The carrier grade telecom infrastructure includes high availability and scalability as one of the main features, and it is required that the changes are reliably delivered to the production site and continuously monitored. This paper aims to systematically review the approaches, tools and challenges that have been reported to have adopted and implemented continuous practices in industries and create a proof of concept for DevOps adoption in telecom industry.**

***Keywords*: Development Operations, Continuous Delivery, Continuous Deployment, Continuous Monitoring.**

1. Introduction

For the last two decades software has become an essential part of the business. The lifecycle of software includes coding, building, deployment, validation and verification of all components, and continuous monitoring. There are number of unique software development life cycle models designed to be practiced during the software development process to ensure its success.

DevOps was introduced as an extension to Agile which focuses on operational aspects. The word DevOps is a combination of two words Development and Operation which signifies the aim of integrating the two worlds. It's an organizational shift which aims for faster, flexible software delivery and services with higher quality and reliability. DevOps extends the agile movements of continuous development goals to continuous integration and release [1].

In industry Continuous Integration (CI) is a widely established practice which enables frequent release cycles and improved software quality. Frequent communication is established among peers and changes are regularly committed to a single source repository with automated builds and testing. [2].

The Continuous Delivery (CDE) aims to ensure that an application is always production ready. It provides benefits such as faster feedback and reduction of cost and deployment risks [3]. Continuous Deployment (CD), on the other hand, aims to continuously deploy changes to the software in the production environment [4], [5]. Continuous Integration, Delivery and Deployment could be referred as pillars on which a successful implementation of DevOps is dependent on.

Traditional telecom and IT infrastructure was mainly hardware oriented and has now been increasingly moving towards virtualized environments. The carrier-grade telecom infrastructures require high availability, scalability and the ability to monitor performance parameters. Usually the deployment is done in production environment manually and is then manually tested once the development is complete which leads to longer periods of waiting. With most of the deployment being done manually, there is an increased chance of manual error. Adoption of automation of continuous practices in DevOps culture using open source tools is expected to provide benefits of: (1) frequent releases which will improve product quality and lead to customer satisfaction (2) establishing a quick feedback cycle (3) elimination of manual tasks (4) Continuous validation and monitoring in production environment (5) Reduced licensing cost. This paper aims to review the approaches, tools and challenges that have been reported in implementing the continuous practices in in industries and make a case for adoption of these practices in telecom industries.

This paper is structured as follows. First, we introduce the challenges faced in adoption of continuous practices of DevOps in various industries and map them possible solutions in Section 2. In Section 3, we discuss various tools that can be employed to design and implement a DevOps Pipeline for telecom industry based on review of related work. In Section 4, we present our conclusions and ideas for future work.

2. Common challenges to adopt DevOps practices in telecom industry

Under this section, the challenges of implementing DevOps in telecom industry are listed:

1. *Lack of Awareness:* DevOps is relatively a new concept and many organizations are still trying to understand and establish what it means and what impact it will have on the business. [9] There is a lack of understanding on the transformations that must be done to adapt DevOps into industries [10]. To overcome this challenge, knowledge sharing between teams on different levels has to be held. If required, training should be provided from leading industry influencers [6].
2. *Communication Challenges:* Successful implementation of continuous practices in DevOps requires more coordination and collaborations among team members. Brown and Starkey [7] state that communication directly impacts decisions made by an organization. There exists a communication gap between developers and operations which leads to a pressing need for strengthening the harmonization of an IT organization [8].
3. *Lack of Investment:* For any change to be implemented, cost factor plays a main role. L. Chen [11] found that hardware dependency and compatibility challenges were faced while automatically deploying changes into customer environment which will require system upgradation. These upgrades are associated with high costs. Total investment will also include cost of training the team members to improve qualification and expertise for adoption of continuous practices.
4. *General Resistance to change:* Regardless of how inefficient they may be people tend to prefer working with long established processes. This resistance towards change may sabotage new ideas and thoughts [9]. According to researchers, communicating how the change will benefit the organization and the employees individually might help to deal with resistance [8].

3. Related work

A brief literature survey was carried out to gain understanding about various concepts like continuous practices and tools used for the purpose. Excerpts from a few of the papers/ journal referred are as follow:

The existing processes in carrier grade telecom infrastructures are mainly designed for preplanned, hardware-oriented, physical infrastructures. [12]. The shift of these infrastructures to software defined and virtualized networks has created a need for more elastic, programmable, dynamically optimizing and scalable environments [13].

Mojtaba Shahin, et. al. [14] provide a detailed systematic review of continuous practices being adopted in various industries. They have observed that there is an increased interest in past few years and is spread across wide range of application domains.

The Continuous Integration tools, Git for version control and Jenkins as Integration Server are used in deployment pipelines in [15] and [16]. Automation of the Continuous Integration can be achieved in Jenkins which can also trigger acceptance tests [17]. The reports are stored in Jenkins servers for reference.

The process of continuous deployment in virtual cloud environments require tools for automated deployment. Christian Makaya and Douglas Freimuth [18] proposed VNF Onboarding Automation Tool(VOAT) provides a framework for automated deployments of VNFs in public or private cloud environments. It maps available metadata to the artifacts and pushes them to necessary production systems to complete VNF onboarding. T-NOVA [19] is yet another framework proposed by G. Xilouris et. al., which allows service providers to deploy network functions and also provide value added services like monitoring and optimization. Chef [21], an open source software has been used for deployment automation in [20].

In DevOps scenarios the frequent releases and automated deployments require automated continuous releases as well. Michael Wurster et al., [22] propose a concept where developers define test along with deployment model. These tests are run at runtime after successful deployment of VNFs. Functionality of carrier grade products can be tested with automated regression suites created using various library extensions of Robot framework where everything is checked out automatically and detailed reports of outcomes are published [23].

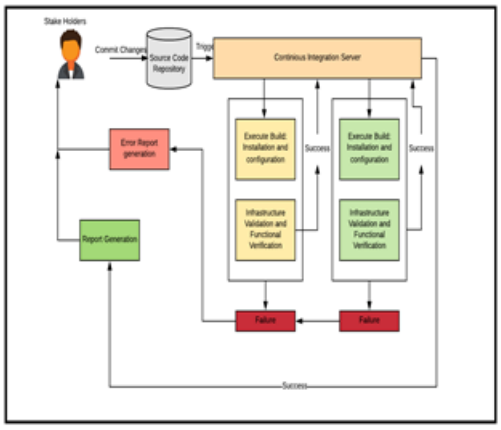

Fig. 1. Deployment pipeline

4. Deployment pipeline for carrier grade telecom products

The telecom carrier grade products are increasingly moving to virtualized cloud environments where network functions are no longer static and embedded into physical boxes deployed at fixed points. This makes it more dynamic, innovation enabling and efficient.

A deployment pipeline is a way to deal with software development process by breaking it up into stages which are chained together. Every commit to the software goes through a complex process on its way to being released. That process involves building the software, followed by the progress of

these builds through multiple stages of testing and deployment. The requirement to deploy a software in continuous basis at the production environment has increased the need for deployment pipeline. Also, the success of adopting the continuous processes in DevOps is heavily dependent on them. [24]. Fig. 1 illustrates the main components of deployment pipeline.

The given proof of concept requires the following configurations:

- Integration of Jenkins server to the Git source code repository which contains source code and test directory.
- Configure the Jenkins to automatically build, deploy and test when a commit is made to the repository.
- Integrate Jenkins with Static code analysis tool, Binary Repository and Configuration management tool.

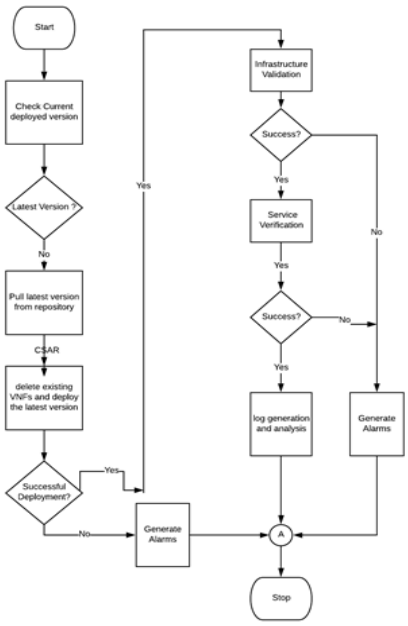

Fig. 2. Flow diagram

Teamwork is essential in software engineering and software engineers need to be equipped with the newest tools, techniques and skills to work in teams. With many developers working on the same project there is higher probability that temporary code changes clash with other developer's code. Version control tools enable developers to simultaneously work on the same project. Developers can edit local copies of the files and chose when to share with rest of the team.

Continuous integration is a process where developers commit their code regularly to a centralized repository which triggers a build followed by acceptance testing. Jenkins, a popular java based open source CI tool which can be triggered on every commit made to Git repository, is used to run cloud deployment and acceptance tests. The test reports are stored in Jenkins server and mailed to respective stakeholders. Each new build is deployed in test environment and infrastructure validation and functionality verification is of all components is carried out. The changes are then deployed in production environment. Here, again the verification and validation are carried out. The flow diagram in Fig. 2, represents the stages of deployment and testing that is carried out in a test or production environment.

5. Conclusion and future work

This paper throws light on how automation can help reduce manual tasks. The proof of concept demonstrates various benefits of employing end to end automation in telecom carrier grade infrastructures that are cloud based.

The future work needs to be focused on security and monitoring of application performance.

References

[1] M. Virmani, "Understanding DevOps & bridging the gap from continuous integration to continuous delivery," Fifth International Conference on the Innovative Computing Technology (INTECH 2015), Pontevedra, 2015, pp. 78-82.
[2] M. Fowler, "Continuous Integration," http://martinfowler.com/articles/continuousIntegration.html
[3] J. Humble. "Continuous Delivery vs. Continuous Deployment," https://continuousdelivery.com/2010/08/continuous-delivery-vs-continuous-deployment/
[4] I. Weber, S. Nepal and L. Zhu, "Developing Dependable and Secure Cloud Applications," in *IEEE Internet Computing*, vol. 20, no. 3, pp. 74-79, May-June2016.
[5] What is Continuous Deployment, accessed on May. 5, 2019. http://electric-cloud.com/resources/continuous-delivery-101/continuous-deployment/
[6] M. B. Kamuto and J. J. Langerman, "Factors inhibiting the adoption of DevOps in large organisations: South African context," *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, 2017, pp. 48-51.
[7] A. D. Brown, K. Starkey, "The effect of organizational culture on communication and information", Journal of Management Studies-Oxford, vol. 31, pp. 807-807, 1994.
[8] S. W. Hussaini, "Strengthening harmonization of Development (Dev) and Operations (Ops) silos in IT environment through systems approach," *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Qingdao, 2014, pp. 178-183.
[9] L. F. Wurster, R. J. Colville, C. Haight, S. Tripathi, and A. Rastogi, "Emerging technology analysis: DevOps a culture shift, not a technology," Gartner, 2013. https://www.gartner.com/doc/2571419/emerging-technology-analysisdevops-culture.
[10] L. Zhu, L. Bass, and G. Champlin-Scharff, "DevOps and its practices," IEEE Software, vol. 33, no. 3, pp. 32–34, May 2016.
[11] L. Chen, ``Towards architecting for continuous delivery," in Proc. 12th Work. IEEE/IFIP Conf. Softw. Archit. (WICSA), May 2015, pp. 131-134.
[12] W. John et al., "Service Provider DevOps," in IEEE Communications Magazine, vol.55, no.1, pp. 204-211, January 2017.
[13] C. Meirosu et al., "DevOps for Software-Defined Telecom Infrastructures," Internet Draft, draft-unify-nfvrg-devops (work in progress), July 2016.
[14] M. Shahin, M. Ali Babar and L. Zhu, "Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices," in IEEE Access, vol. 5, pp. 3909-3943, 2017.
[15] Martin Brandtner, Emanuel Giger, Harald Gall, SQA-Mashup, Information and Software Technology, v.65 n.C, p.97-113, September 2015
[16] Steve Neely, Steve Stolt, Continuous Delivery? Easy! Just Change Everything (Well, Maybe It Is Not That Easy), Proceedings of the 2013 Agile Conference, p.121-128, August 05-09, 2013
[17] N. Seth and R. Khare, "ACI (automated Continuous Integration) using Jenkins: Key for successful embedded Software development," 2015 2nd

International Conference on Recent Advances in Engineering & Computational Sciences (RAECS), Chandigarh, 2015, pp. 1-6
[18] C. Makaya and D. Freimuth, "Automated virtual network functions onboarding," 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Palo Alto, CA, 2016, pp. 206-211.
[19] G. Xilouris et. al., "T-NOVA: A marketplace for virtualized network functions," 2014 European Conference on Networks and Communications (EuCNC), Bologna, 2014, pp. 1-5.
[20] E. Luchian, C. Filip, A. B. Rus, I. Ivanciu and V. Dobrota, "Automation of the infrastructure and services for an openstack deployment using chef tool," 2016 15th RoEduNet Conference: Networking in Education and Research, Bucharest, 2016, pp. 1-5.
[21] Chef [Online]: https://www.chef.io/
[22] M. Wurster, U. Breitenbücher, O. Kopp and F. Leymann, "Modeling and Automated Execution of Application Deployment Tests," 2018 IEEE 22nd International Enterprise Distributed Object Computing Conference (EDOC), Stockholm, 2018, pp. 171-180.
[23] Q. Na and D. Huaichang, "Extension and application based on robot testing framework," 2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, 2015, pp. 428-431.
[24] A. Phillips, M. Sens, A. de Jonge, and M. van Holsteijn, The IT Man ager's Guide to Continuous Delivery: Delivering Business Value in Hours, XebiaLabs, Hilversum, The Netherlands, 2015.