ISSN- 2394-5125 VOL 7, ISSUE 2, 2020

DevOpsand Selection of Appropriate Tool: A Review

Dr. Mayank R. Kothawade
Asst. Professor, VIIT Baramati, Savitribai Phule Pune University

Abstract: DevOps has exploded in popularity since the concept first emerged in 2009. A DevOps approach encompasses both practices and tools that engineers use to build quality software more rapidly. Before DevOps, software development followed a "waterfall" model. Back then, developers would code first, then perform quality assurance testing, and fix any bugs or errors as necessary. The DevOps model is much more integrated; it involves breaking down the development process into smaller parts, which are easier and faster to manage. Instead of building the entire application before testing, developers build and test continuously. They build the entire model on a continuous improvement process involving daily collaboration. DevOps teams accomplish this with automation tools, from deployment and continuous integration to monitoring, security, and cost management. In this research paper brief overview of DevOps is covered as well as various crucial DevOps tools were categorized and their analysis withstrength, weakness, opportunities and threat they carry.

Keywords: DevOps, DevOps tools, SCRUM, Kanban, Agile, Docker, Jenken, DevOps phases.

Objectives:

1. To understand DevOps.
2. To identify and classify the important DevOps tools.
3. To understand strength, weaknesses, opportunities and threats of selected DevOps tools.

1. Introduction

DevOps is a culture that encourages collaborations among all stakeholders including development and operations teams and the improvement of processes through automation to increase the quality and speed of software delivery.The goal is to create a working environment in which building, testing and deploying software can occur quickly, reliably and frequently. By doing this, a company can achieve its goals faster and deploy new features, security patches, and bug fixes[1]. The most recent trends across the industry related to DevOps are as follows:

- End-to-end lifecycle management to streamline DevOps workflows
- Changing the definition of an "application"
- Moving fast without breaking things
- Optimizing the DevOps toolchain
- Closing the developer skill gap
- DevSecOps is reality

ISSN- 2394-5125 VOL 7, ISSUE 2, 2020

- Container-first strategy fully adopted

2. What is DevOps?

DevOps is an evolution of the agile movement [2]. Agile Software Development advocates smallrelease iterations with customer reviews. It assumes the team can release software frequentlyin some production-like environment. However, pioneer Agile literature puts little emphasis ondeployment-specific practices. No Extreme Programming (XP) practice, for example, is aboutdeployment [3]. The same sparse attention to deployment is evident in traditional softwareengineering processes [4] and books [5, 6]. Consequently, the transition to production tendsto be a stressful process in organizations, containing manual, error-prone activities, and, even,last-minute corrections [8]. DevOps proposes a complementary set of agile practices to enable theiterative delivery of software in short cycles effectively.

From an organizational perspective, the DevOps movement promotes closer collaboration between developers and operators. The existence of distinct silos for operations and development isstill prevalent: operations staff are responsible for managing software modifications in productionand for service levels [9]; development teams, on the other hand, are accountable for continuouslydeveloping new features to meet business requirements. Each one of these departments has itsindependent processes, tools, and knowledge bases. The interface between them in the pre-DevOpsera was usually a ticket system: development teams demanded the deployment of new softwareversions, and the operations staff manually managed those tickets.In such an arrangement, development teams continuously seek to push new versions intoproduction, while operations staff attempt to block these changes to maintain software stability andother non-functional concerns.

Theoretically, this structure provides higher stability for softwarein production. However, in practice, it also results in long delays between code updates anddeployment, as well as ineffective problem-solving processes, in which organizational silos blameeach other for production problems.Conflicts between developers and operators, significant deployment times, and the need forfrequent and reliable releases led to inefficient execution of agile processes. In this context, developers and operators began collaborating within enterprises to address this gap. This movementwas coined “DevOps” in 2008 [10].In the Continuous Delivery book [11], Humble advocates for an automated deployment pipeline,in which any software version committed to the repository must be a production-candidate version.After passing through stages, such as compilation and automated tests, the software is sent toproduction by the press of a button. This process is called Continuous Delivery. A variant is thecontinuous deployment [12], which automatically sends to production every version that passesthrough the pipeline.

ISSN- 2394-5125 VOL 7, ISSUE 2, 2020

2.1 DevOps methods

There are a few common DevOps methods that organizations can use to speed and improve development and product releases. They take the form of software development methodologies and practices. Among the most popular ones are Scrum, Kanban, and Agile:

- **Scrum**: Scrum defines how members of a team should work together to accelerate development and QA projects. Scrum practices include key workflows and specific terminology (sprints, time boxes, daily scrum [meeting]), and designated roles (Scrum Master, product owner).
- **Kanban:** Kanban originated from efficiencies gained on the Toyota factory floor. Kanban prescribes that the state of software project work in progress (WIP) be tracked on a Kanban board.
- **Agile:** Earlier agile software development methods continue to heavily influence DevOps practices and tools. Many DevOps methods, including Scrum and Kanban, incorporate elements of agile programming. Some agile practices are associated with greater responsiveness to changing needs and requirements, documenting requirements as user stories, performing daily standups, and incorporating continuous customer feedback. Agile also prescribes shorter software development lifecycles instead of lengthy, traditional "waterfall" development methods.

3. DevOps tools

DevOps Tool is an application that helps automate the software development process. It mainly focuses on communication and collaboration between product management, software development, and operations professionals. DevOps tool also enables teams to automate most of the software development processes like build, conflict management, dependency management, deployment, etc. and helps reduce manual efforts [13].These tools facilitate ways for effective sharing and exchange of information and technical know-how between all stakeholders be it development, operations, security or business teams for effective product output [14, 15].There are more than 300 plus tools used during the software life cycle phases, but this research paper highlights upon few important and popular tools. Table 1 shows the category wise list of DevOps tools whereas table 2 shows the categorization of DevOps tools as per its life cycle phases.

ISSN- 2394-5125 VOL 7, ISSUE 2, 2020

Table 1: Category wise DevOps tools

Sr.	Category	DevOps Tool
1	Configuration Management	Chef, Puppet, Ansible, SaltStack, AWS Systems Manager
2	CI/CD	CircleCI , GitLab, Jenkins, Semaphore, CloudBees, AWS CodePipeline, AWS CodeDeploy, GitHub, Bamboo, IBM UrbanCode
3	Continuous Monitoring/ Observability	HoneyComb Observability, Epsagon, DataDog, SignalFX, New Relic, Graphana, Dynatrace,AWS CloudWatch, Sensu
4	Log Management	Sumo Logic, Splunk, ChaosSearch, Logz.io, Loggly, AWS CloudWatch,
5	Containers and Kubernetes	Docker, Kubernetes, CloudZero Kubernetes Cost Monitoring and Analysis, Rancher Labs, Fairwinds, AWS ECS, AWS EKS, ArKade, Mesos
6	Incident Response and Management	PagerDuty, Splunk On-Call,
7	Infrastructure As Code	Terraform by HashiCorp, AWS CloudFormation,
8	Chaos Engineering	Gremlin, Chaos Monkey, Chaos Toolkit
9	Security	Lacework, Snyk, Threat Stack, Fugue, ecure Code Warrior
10	Cloud Cost Optimization and Intelligence s	AWS Cost Explorer, AWS Cost Anomaly Detection, Spot.io, Opsani, ProsperOps, Kubecost, GitLab, Replex.io, CloudZero
11	Version Control	GitHub, Bitbucket, GitLab
12	Application Performance Management	Prometheus, Dynatrace, AppDynamics
13	Test automation tools	Test.ai, Ranorex, Selenium
14	Artifact Management	Sonatype NEXUS, JFRog Artifactory, CloudRepo,
15	Codeless Test Automation tools	AccelQ, Appvance, Testim.io
16	Collaboration	Jira, HipChat, Slack, Trello

ISSN- 2394-5125 VOL 7, ISSUE 2, 2020

Table 2: DevOps tools as per life cycle phases

Sr.	Phase	DevOps Tool
1	Continuous Development	Git, SVN, CVS, Mercurial, Maven, Gradle
2	Continuous Testing	Jenkins, TeamCity, Travis
3	Continuous Integration	Jenkins, Selenium TestNG, JUnit
4	Continuous Deployment	Configuration Management Tools – Chef, Puppet, Ansible and Containerization – Docker, Vagrant
5	Continuous Monitoring	Splunk, ELK Stack, Nagios, New Relic

4. Analysis of important tools w.r.t. strength, weakness, opportunities and threat

When you adopt a DevOps approach to building and operating software systems, you must rely on modern tools for almost every aspect of build, release, and operations activities. But before you get into the weeds of comparing one tool against another, you need to think more broadly about what you need[16].Perform a strength, weakness, opportunities and threat analysis on your business. Define strengths, weaknesses, opportunities and threats. Expand on the analysis with a strategic plan to address each area. How do you protect or ensure strengths and opportunities while mitigating risks and threats?After reading more about DevOps, consider where it fits with your organization. What principles and practices would you consider useful to adopt? Why?

Also consider the Agile and Waterfall models as well as Scrum. Again, which would you adopt? Will more than one work for different aspects of the business? Create a diagram or flowchart detailing how the model or framework you chose would work within your structure. Address each of these areas and submit as a Word document with any required tables, diagrams, or images embedded.

This section gives the analysis of key tools as listed below;

- GitLab
- Chep
- Puppet
- Windows System Center Configuration Manager (SCCM)
- MACs JAMF
- Docker
- Airwatch

ISSN- 2394-5125 VOL 7, ISSUE 2, 2020

GitLab

Strength

- Complete DevOps platform with seamless connections
- Industry leading source code management, code review, and CI products
- Ability to start with GitLab just on the use case needed.
- Strong analyst relationships
- Expansive feature set and roadmap
- Open core with both self-managed and SaaS options

Weaksnesses

- Not yet enterprise grade, and several improvements are needed for enterprise adoption of self-managed instances.
- Performance issues continue to mount in various areas
- Some areas of the product are not at viable or complete maturity

Opportunities

- Consideration of new business models
- Big opportunity to grow and/or shift usage to SaaS
- Creating an "easy migration" path from other popular DevOps tools
- Capitalize on solutions that can tie together information
- Focus on internal adoption of the features

Threats

- Onboarding new team members may lead to a slow down in velocity if not managed carefully.
- Continious innovations and imporvements by the competetiors
- DevOps tools start to become commoditized, with IaaS providers giving them away for free with IaaS/cloud

Chep

Strength

- Mature
- Scalability
- Less expensive
- Version Control
- Cloud Tested
- Full stack
- AWS

Weaksnesses

- Medium learning curve
- Easy to lose convergence
- Secrets management

Opportunities

- Build once and use any time when required
- Cost saveyy
- Tribal knowledge probelm made obsolete

Threats

- Expertise difficult to find
- Complexity creep
- Credentials Safe-Keeping
- Containers

ISSN- 2394-5125 VOL 7, ISSUE 2, 2020

Puppet

Strength

- Mature
- Granular control
- Less expensive
- Version Control
- Full stack
- Low learning curve
- AWS

Opportunities

- Build once and use any time when required
- Cost saveyy
- Tribal knowledge probelm made obsolete

Weaksnesses

- AWS, On-prem hybrid not possible or not mature
- Scalability
- Secrets management

Threats

- It may become legacy
- Complexity creep
- Credentails safe keeping
- Containers

Windows System Center Configuration Manager (SCCM)

Strength

- Full stack
- Great support
- Wide integration
- Patching
- Granular control
- Low learning curve
- Seamless AD integration

Opportunities

- Build once and use any time when required
- Cost saveyy
- Tribal knowledge probelm made obsolete

Weaksnesses

- Expensive
- GUI usage
- AWS, On-prem hybrid will require Azure

Threats

- Possibility of cost increase
- Windows-DevOps-SCCM skill set may become a orgnization weakness
- Maintaining both SCCM and Chef may prove to be costly and risky for support

ISSN- 2394-5125 VOL 7, ISSUE 2, 2020

Strength

- Less expensive
- Up-to-date with OSX releases
- Great support for MDM
- Inventory Management
- Central Deployment
- Backend is compatible to windows

Opportunities

- Safe bet

MACs
JAMF

Weaksnesses

- Patching
- Steep learning curve
- Best practices not documented

Threats

- Losing market share to Airwatch in mobile space

Strength

- A strong parent brand to back it up
- Financial Strength is more and hence can experiment
- Strong Distribution System
- Innovation is the Point of Difference
- Globally recognized and avilable

Opportunities

- Fast growing casual wear
- Changing cutomer behavior and experimental generation
- Increasing disposable income and awareness of fashion trends

Docker

Weaksnesses

- Brand dilution due to shift of focus from 25-49 to the youth
- Pants market is a niche market which keeps the potential market small

Threats

- Competitive Market due to international players
- Product substitute available
- Fake imitations and replica products affect brand image

ISSN- 2394-5125 VOL 7, ISSUE 2, 2020

Strength
- Simplicity
- Great MDM support
- Secure content locking
- Mail integration + Exchange
- Superior application management
- High availability

Opportunities
- Maerket leader in MDM
- Feels like next thing
- Owned by VMware

Airwatch

Weaksnesses
- Installation is complex
- Feature overhead
- High cost

Threats
- People with required skillset may become scarce

Conclusion

The software industry is adapting DevOps at a rapid pace. Enterprises are eager to take the benefits of quicker application development, testing, and deploying at a high rate of innovation and with a more steady operating environment. DevOps is making teams more flexible and productive by providing several tools for different phases and operations. While selecting an appropriate tool or multiple tools it becomes necessary to have proper insights of that tool. This research paper covers the thorough analysis of few important tools which helps for proper tool selection. Only the limitation with his paper is that it does not cover the analysis of all available tools.

References

1. Trending DevOps Topics in 2020, Keep up with the latest DevOps trends to finish the year out stronghttps://levelup.gitconnected.com/7-trending-devops-topics-in-2020-155078d6f550, Retrieved 16December 2019
2. Henrik Bærbak Christensen. 2016. Teaching DevOps and Cloud Computing Using a Cognitive Apprenticeship andStory-Telling Approach. In Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer ScienceEducation (ITiCSE ’16). ACM, 174–179. Code: A47
3. Kent Beck and Cynthia Andres. 2004. Extreme programming explained: embrace change. Addison-Wesley Professional.
4. Per Kroll and Philippe Kruchten. 2003. The rational unified process made easy: a practitioner’s guide to the RUP. Addison-Wesley Professional.

ISSN- 2394-5125 VOL 7, ISSUE 2, 2020

5. Roger S Pressman. 2005. Software engineering: a practitioner's approach (6th ed.). Palgrave Macmillan.
6. Ian Sommerville. 2011. Software engineering (9th ed.). Addison-Wesley.
7. Jez Humble and David Farley. 2010. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment
8. Automation. Addison-Wesley Professional.
9. Eoin Woods. 2016. Operational: The Forgotten Architectural View. IEEE Software 33, 3 (2016), 20–23. Code: A82
10. Patrick Debois. 2008. Agile Infrastructure & Operations. (2008). At Agile 2008 Toronto. Slides available onhttp://www.jedi.be/presentations/agile-infrastructure-agile-2008.pdf, Retrieved 20 September 2019.
11. Jez Humble and David Farley. 2010. Continuous Delivery: Reliable Software Releases through Build, Test, and DeploymentAutomation. Addison-Wesley Professional
12. Jez Humble. 2010. Continuous Delivery Vs Continuous Deployment. (2010). https://continuousdelivery.com/2010/08/continuous-delivery-vs-continuous-deployment/, Retrieved 24 September 2019.
13. Understanding DevOps Tools – Development, Testing & Deployment Technologies Involved in DevOps, https://www.edureka.co/blog/devops-tools, Retrieved 24 September 2019
14. What is DevOps?https://www.netapp.com/devops-solutions/what-is-devops/, Retrieved 24 September 2019
15. DevOps Tools SWOT analysis, https://www.slideshare.net/mamunrashid001/devops-tools-swot-analysis, Retrieved24 September 2019