

DevOps and Tools Used: A Systematic Review

Kumar Aditya Raj¹, Abhinav Anand², Sahana V³

1,2 (Student, Department of ISE, JSS Academy of Technical Education, Bangalore, India)

3 (Assistant Professor, Department of ISE, JSS Academy of Technical Education, Bangalore, India)

Abstract: DevOps originates from two words development and operations. In the development process, we develop the software and in the operations, we deploy and monitor the systems. DevOps is a methodology that allows a single team to manage the entire application development life cycle that is development, testing, automating, deployment, and operations and monitoring. The objective of DevOps is to shorten the software development life cycle. It is a software development approach through which superior quality software can be developed quickly and with more reliability. There are many tools like Git, Chef, Ansible, Docker, Kubernetes, and many more that help us in the process of implementing the DevOps practice in the industry. This paper is about what DevOps is and dives into the tools as to how and for what they are used. We will also see some surveys about DevOps like the percentage of companies using it and many more [1].

Keywords – Ansible, DevOps, Docker, Git, Kubernetes

I. Introduction

DevOps means Development and Operations which is building the product and maintaining it to run successfully on servers and monitoring them. In earlier days, the developers and operations teams were different and they used to get a lot of issues in coordinating with each other therefore there was a need for a new process to develop software and maintain it. In DevOps, both the development and operations teams work together to develop and ship new features in the existing software and maintain it [4]. Earlier it was adopted by fewer companies as they do not have a large code base or they were not fully convinced that it would be great to shift to the agile method of developing software. But, today's scenario where every company is in a race to ship new features daily has adopted this DevOps practice in which it can ship new features at a light speed and can monitor the systems whether they are performing well or not. There are many tools to implement the agile DevOps lifecycle. To know about these tools first let us have a look at what are the different stages in DevOps and then we will deep dive into the tools used. First comes the development of a product or software in which the developers build the project from scratch after that they build it and test it for quality and assurance then the product is deployed on the servers after that it is monitored for bugs and issues. So now comes the tools that are mostly used Git is the first tool which is used for version control, Jenkins is used for automation, Chef and Ansible are used for configuration management, Docker and Kubernetes are used for containerization and orchestration, and tools, like terraform, are used for monitoring [2].

II. DevOps tools

2.1. CHEF

Configuration management tool comes under operations and is of two types push-based and pull-based [6]. Push configuration server pushes the configuration to the nodes. E.g: - ANSIBLE. Pull configuration nodes check with the server periodically and fetch the configuration from it. E.g: - CHEF and PUPPET written in ruby and erlang. A chef is an administration tool. Whatever system admin used to do manually now we are automating all those tasks by using chef. Configuration management is a method through which we automate the admin tasks. Configuration management tool turns your code into infrastructure So your code would be repeatable, testable, and versionable.

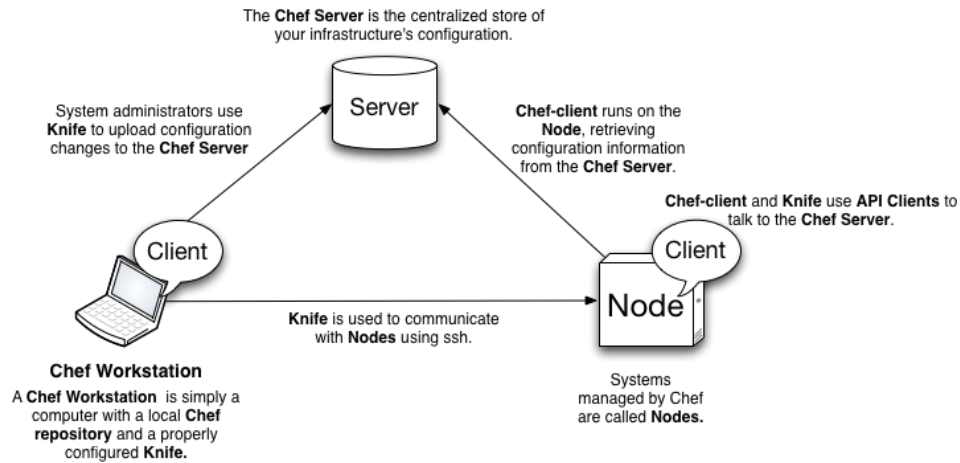


Fig1.Workingofchef

2.1.1. COMPONENTS OF CHEF

Workstations are personal computers or virtual servers where all configuration code is created tested or changed. DevOps engineers sit here and write codes. This code is called a recipe. A collection of recipes is known as a cookbook. The workstation communicates with the chef server using a knife. A knife is a command-line tool that uploads the cookbook to the server. The chef server is a middleman between the workstations and the nodes. All cookbooks are stored here. The server may be hosted locally or remotely. Nodes are the systems that require configuration. Ohai fetches the current state of the node is located. Node communicates with the chef server using the chef-client. Each node can have a different configuration required Chef client is installed on every node.

2.2. ANSIBLE

Ansible is a push-based configuration management tool. It uses YAML(Yet Another Markup Language) scripting. YAML works on key pairs. ANSIBLE is an open-source IT configuration management, deployment, and orchestration tool. It helps in many automation challenges of code and decreases the delivery time of the team.

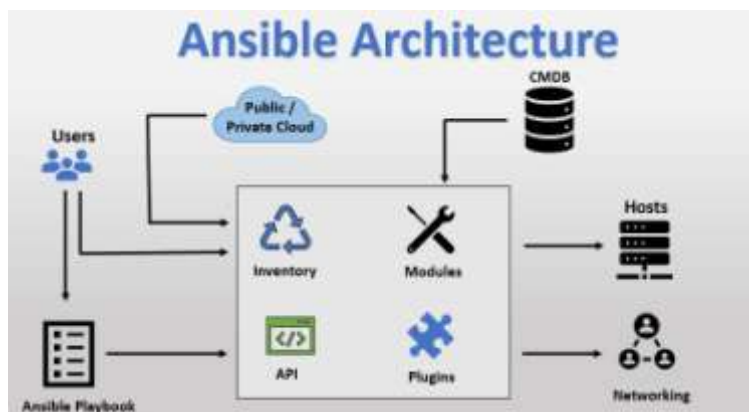


Fig2.Ansiblearchitecture

2.2.1. TERMS USED IN ANSIBLE

Ansible Server, the machine where Ansible is installed and from which all tasks and playbooks will be run. A module is a command or set of similar commands meant to be executed on the client-side. Task - A task is a section that consists of a single procedure to be completed. The role, A way of organizing tasks and related files to be later called in a playbook. Fact - information fetched from the client system from the global variables with the gather-facts operation. Inventory - File containing data about the ansible client servers. Play - execution

of a playbook. Handler - a task that is called only if a notifier is present. Notifier - section attributed to a task that calls a handler if the output is changed. Playbooks - it consists of code in YAML format, which describes tasks to be executed. Host - Nodes, which are automated by Ansible. CI / CD JENKINS, MAVEN, NAGIOS. CI means continuous integration and CD means continuous deployment or continuous delivery.

CHEF AND ANSIBLE COMPARISON

Criteria	ANSIBLE	CHEF
ConfigurationLanguage	YAML,Python	Ruby,Json
Architecture	Agentless(ClientOnly)	AgentBased(Client-Server)
DeploymentMethod	PushModel	PullModel
FilesCreatedBeforeOperation	Playbooks	Recipe
TransportMechanism	SSH	REST
Portused	TCP port22	TCP port10002

Table1.Comparison betweenChef andAnsible

2.3. DOCKER

Docker is used for containerization. A container is like a virtual machine. Docker is an advanced version of virtualization i.e containerization. Docker is an open-source centralized platform designed to create, deploy and run applications. Docker uses containers on the host operating system to run applications. It allows applications to use the same kernel as a system on the host computer rather than creating a whole virtual OS. We can install docker on any OS but docker-engine runs natively on Linux distribution. Docker is written in the go language. Before docker, many users faced the problem that a particular code is running in the developer’s system but not in the user’s system. Docker is a set of platforms that used OS level virtualization whereas VMWare used hardware-level virtualization [9] .

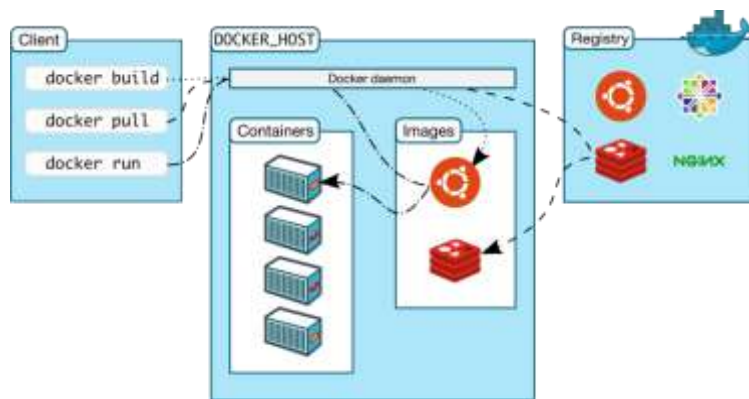


Fig3.Workingofdockerizedapplication

2.3.1. ADVANTAGES OF DOCKER

No preallocation of ram. CI efficiency - docker enables us to build a container image and use that same image across every step of the deployment process. Less cost It is light in weight. It can run on physical hardware or virtual hardware or the cloud You can reuse the image. It take very less time to create the container.

2.4. JENKINS

Jenkins is an open-source project written in Java that runs on windows, macOS, and other UNIX-like operating systems. It is free, community-supported, and might be your first choice tool for ci. Jenkins automates the entire software development life cycle. Jenkins was originally developed by sun microsystem in 2004 under the name Hudson. The project was later named Jenkins when oracle bought microsystems It can run on any major platform without any compatibility issues. Whenever developers write code we integrate all that code of all developers then and we build, test, and deliver/deploy it to the client. This process is called ci/cd. Jenkins helps us to achieve this. Because of ci, new bugs will be reported fast and get rectified fast so the entire software development happens fast.

2.4.1. WORKFLOW OF JENKINS

We can attach git, Maven, selenium, and artifactory plugins to Jenkins. Once developers put code in Github, Jenkins pulls that code and sends it to maven for build once the build is done Jenkins pulls that code and sent to selenium for testing. Once testing is done, then Jenkins will pull that code and send it to artifactory as per requirement and so on. We can also deploy with Jenkins.

2.5. KUBERNETES

Kubernetes is an open-source container management tool that automates container deployment, container scaling, and load balancing. It schedules, runs and manages isolated containers that are running on virtual / physical/cloud machines. All top cloud providers support Kubernetes [10] .

2.5.1. FEATURES OF KUBERNETES

- Orchestration (clustering of any no of containers running on different n/w) [11]
- Auto-scaling (vertical and horizontal)
- Auto healing
- Load balancing
- Platform independent(cloud, virtual, physical)
- Fault tolerance(node, pod failure)
- Rollback (going back to the previous version)
- Health monitoring of containers

DOCKER AND KUBERNETES COMPARISON

FEATURES	KUBERNETES	DOCKER
InstallationandClusterConfiguration	Installationiscomplicatedbutclusterisstrong	Installation is simple but cluster is notstrong
GUI	GUIisKubernetesdashboard	NoGUI
Scalability	Highlyscalable&scalesfast	ScalesfasterthanKubernetes

Auto-scaling	Kubernetescandoautoscaling	Dockercannotautoscale
LoadBalancing	Manual intervention needed for loadbalancingbetweendifferentcontainers	Docker auto balances between thecontainers
RollingUpdates&Rollbacks	Can deploy rolling updates and doesautomaticrollbacks	Can deploy rolling updates, but notautomaticrollbacks
DataVolumes	Can share storage volumes only withothercontainersinsamepod	Can share storage volumes with anyothercontainer
LoggingandMonitoring	In-builttoolsforloggingandmonitoring	Thirdpartytoolsforloggingandmonitoring

Table2.Comparison between Docker and Kubernetes

III. Analysis Of Market Trend

Over the last 15 years, tens of thousands of organizations have adopted a DevOps way of working with the help of various tools. It has been seen how DevOps has grown from a term only familiar to technical teams to becoming part of the C-suite vocabulary. Practices like CI/CD and automation have become the norm in every engineering organization. Here are some insights from a survey done by Atlassian in 2020.



Fig 4. Impact of DevOps



Fig5.Issues with DevOps



Fig6.Impact survey data

IV. Conclusion

In summary, this paper has analysed various DevOps tools and the emergence of DevOps as a substantial block in companies. The survey done by Atlassian showed how more companies are adopting DevOps based pipelines for the software development life-cycle. DevOps has made the deployment and future upgradation flawless. This helps companies to avoid a lot of manual effort as well as save on software maintenance expenditure. In future, we intend to do a survey to compare the growth and impact of DevOps in software development domain. We would also like to cover more effective though lesser used tools that may have a greater impact on the software industry in the future.

References

- [1]. Floris Rrich, Chintan Amrit, A Qualitative Study of DevOps Usage in Practice, Journal of Software: Evolution and Process. 00. 10.1002/smr.1885
- [2]. Sabharwal, Navin & Rathore, Raminder & Agrawal, Udit. (2022). Introduction to Agile and DevOps 10.1007/978-1-4842-7505-4_3.
- [3]. Kratzke, Nane. (2021). DevOps. 10.3139/9783446472846.003.
- [4]. Liang, Qiao. (2021). DevOps Driven by Developers. 10.1201/9781003221579-16.
- [5]. Singh, Raju. (2020). DevOPS Now and Then. 10.20944/preprints202011 0410.v1.
- [6]. Lokaadinugroho, Indrabudhi & Burhanudin, Burhanudin. (2022). DevOps CICD in Higher Education. 10.21203/rs.3.rs-1500352/v1.
- [7]. Alenezi, Mamdouh & Zarour, Mohammad & Akour, Mohammad. (2022). Can Artificial Intelligence Transform DevOps?. 10.48550/arXiv.2206.00225.
- [8]. Yarlagadda, Ravi Teja. (2021). DevOps and Its Practices. SSRN Electronic Journal. 9. 2320-2882.
- [9]. Chen, Chao-Chun & Hung, Min-Hsiung & Lai, Kuan-Chou & Lin, Duke. (2021). Docker and Kubernetes. 10.1002/9781119739920.ch5.
- [10]. Schmeling, Benjamin & Dargatz, Maximilian. (2022). Developing on and with Kubernetes. 10.1007/978-1-4842-7942-7_3..
- [11]. Ortega, José Manuel. (2020). DevOps and Containers Security: Security and Monitoring in Docker Containers