

Analysis of CI/CD Application in Kubernetes Architecture

^[1]K.Janani, ^[2]K.Anuhya, ^[3]V.L.Manaswini, ^[4]V.Likitha, ^[5]B.Suneetha, ^[6]T.Vignesh

^[1] Computer Science, Koneru Lakshmaiah Education Foundation, ^[2] Computer Science, Koneru Lakshmaiah Education Foundation, ^[3] Computer Science, Koneru Lakshmaiah Education Foundation, ^[4] Computer Science, Koneru Lakshmaiah Education Foundation, ^[5] Computer Science, Koneru Lakshmaiah Education Foundation, ^[6] Computer Science, Koneru Lakshmaiah Education Foundation ^[1]gurulucky199@gmail.com, ^[2] anuhyakurre@gmail.com, ^[3] lakshmimanaswini.vajje@gmail.com, ^[4] Likithavadlamudi14@gmail.com, ^[5] suneethabulla@gmail.com, ^[6] vigneshthangathurai@gmail.com

Article Info

Page Number: 11091 - 11097

Publication Issue:

Vol 71 No. 4 (2022)

Abstract

Technology is your innovation engine and delivering your application faster helps you win. Automation ensures high quality results. CI/CD using Jenkins ansible enables higher efficiency and increased productivity. Jenkins help organizations accelerate the software development process through automation. Jenkins integrates development life-cycle processes of all kinds and by using plugins, achieves CI it makes the preferred choice of many DevOps teams. Ansible is a potent tool for automating the deployment of the application and the provisioning of the target environment. It is an open-source tool for service employment and a solution for software provisioning, application deployment and configuration management. Ansible is a potent tool for automating the deployment of the application and the provisioning of the target environment. It saves time. It is more productive, secure and reliable

Article History

Article Received: 15 September 2022

Revised: 25 October 2022

Accepted: 14 November 2022

Publication: 21 December 2022

Index Terms— Ansible ,Cloud, Continuous Integration and Continuous Development, Devops, Jenkins, Kubernetes

I. INTRODUCTION

Cloud computing can be referred as storing and accessing the data and software on remote servers located online rather than a computer's hard drive or local server. Internet based computing is another name for cloud computing. The components supporting elements needed for cloud computing .These components often refer to front end, Backend. Depending on the service they offer, businesses utilize the three tiers of cloud computing. Infrastructure, Platform, and Application are the three layers. [1].

CI/CD stands for continuous deployment and integration, although the "CD" can alternatively stand for continuous delivery. Continuous deployment goes a step further than continuous delivery in terms of automating various parts of the CI/CD workflow. Making the deployment of new code simple is the aim of continuous delivery. By automating the deployment stage, continuous deployment aims to enable teams to be "hands-off" in the process[2]. CI/CD is a crucial DevOps activity and the best practice for the Agile approach. The application development process can be sped up thanks to this method, which enables development teams to deliver and deploy apps often. Our software development lifecycle can be more effectively automated, and we can track code updates, new features, potential problem fixes, and more by implementing a CI/CD pipeline.

Software deployment and provisioning tools include Ansible and Jenkins. One of the prerequisites for running the Ansible task using CI tools is that it connects through SSH, remote PowerShell, and other basic methods. There are three ways to execute Ansible: the default mode, which requires Ansible to be setup on the Jenkins master; running Ansible playbooks as shell commands; and running jobs on different Nodes[3].

The main cloud deployment methods are private cloud, public cloud, and hybrid cloud. Cloud computing service models are among them. We leverage standard IT methods, such as DevOps and continuous integration/continuous deployment strategies, to deliver services to clients. Human check requirements are reduced thanks to CI/CD. [2]

Most of the software applications have been rapidly increased , updated and these give demand in business. So, without automation in deployment takes lots of time and we need to deploy whenever we make changes in code. After making changes in the cloud, we can easily automate by pressing a single button.

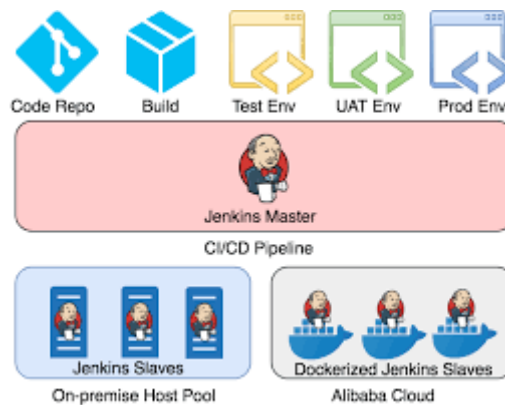
DevOps on AWS is the application of a collection of DevOps cultural ideas, practices, and tools to improve our organization's capacity to create, deliver, and manage apps on AWS at a high level of speed. DevOps allows to handle a single team for entire application lifecycle from testing, deployment, and operations[13]. It helps us to decrease the disconnection between software developer and system administration and it helps to increase speed to deliver application and services. DevOps also allows to serve their customer better and compete more strongly in the market. It is one of the most valuable business disciplines for enterprises or organization

Infrastructure as a Service provides services for computing, storage and networking resources to the end user which helps in saving costs of maintenance and operating prices of servers. It's a complete package for computing. Platform as a Service provides a platform and environment to build and develop applications and services. The services are accessed via internet. Software as a Service provides services to the customers which are hosted in the form of application by the providers. The hosted application can be used by the customers anywhere in the world. Many firms adopt DevOps CI/CD methodologies that satisfy the end user needs and in need to identify the sluggish delivery of services following the Traditional IT practices. As the cloud provides enormous pliability and versatility of the infrastructure, DevOps, which unifies developers and operations teams, may bridge the gap between them. DevOps approaches include Continuous Integration, Delivery and Deployment, which make it easier to automate the entire software delivery process. Automation, in a nutshell, is a representation of the CI/CD central approach. This reduces the need for human inspection and saves time. [5].

II. OVERVIEW OF JENKINS, KUBERNETES AND ANSIBLE

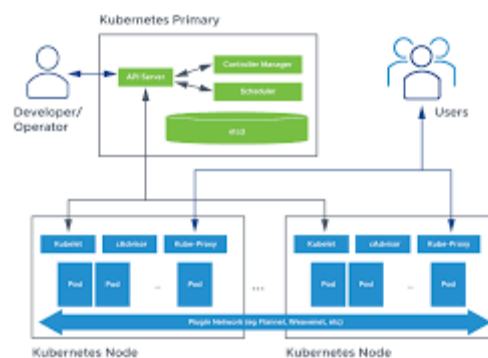
This application is a Java-based, freely available software CI/CD automation solution for DevOps. Pipelines are used to implement CI/CD workflow. The tool simplifies laborious development chores by acting as a "loyal servant." Jenkins enables seamless, continuous creation, testing, and deployment of newly developed code as a continuous integration tool. Jenkins is one of the greatest DevOps tools accessible since it is free, open-source, and flexible, and it can connect to almost any other DevOps tool.

Fig. 1. Jenkins Architecture



For managing workloads and services built on containers, Kubernetes is a portable, adaptable, open-source platform that offers declarative setup and automation. It controls how containerized programmers are running on a cluster of servers and is commonly referred to as "K8s." Using private or public cloud platforms, the K8s solution makes it easier to deploy and manage cloud native apps [3]. Kubernetes automates containerized environments as containers simply need a shared OS kernel to run. Containers are smaller, lighter, and more transportable. The option to deploy containerized apps to Kubernetes clusters, which may run on-premises or in the cloud, has been introduced by application teams. The most widely used container orchestration platform is Kubernetes. As part of the CD stage, software developed in a Git CI pipeline may be sent to Kubernetes. Workloads tied to a GitLab instance may be managed with Kubernetes. It is now a necessary tool for DevOps team and to significantly improve throughput. We got excellent outputs, demonstrating that the potential alternative is dependable, easily scalable, and fast.

Fig. 2. Kubernetes Architecture



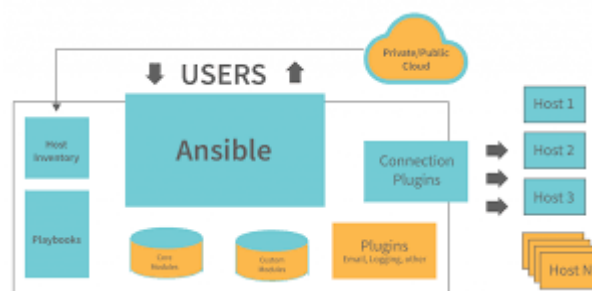
Jenkins is utilized by many DevOps teams because it is a unifies several development lifecycle processes and performs continuous integration with the aid of plugins. Ansible is a potent tool for automating the deployment of the application and the provisioning of the target environment. It is a software provisioning, application deployment, and configuration management solution that is available as an open-source tool for service employment. The creation, integration, and deployment utilizing the pipeline methodology and the ansible tool are the main topics of this work. Time is saved. It is more efficient, safe, and dependable.

The efficiency of the suggested approach, which functions flawlessly even on the least powerful computers provided by the Cloud, was highlighted by using CloudWatch to track resource consumption in AWS. The methods and procedures outlined in this article are an excellent place to start when hosting the fundamental components of the 5G network in Docker containers under

Kubernetes management. In turn, this permits the development of network slices that aid in ensuring the optimum service for various applications, including augmented reality, smart cities, and autonomous vehicles. The deployment flow is continuous and error-free. It is the ideal automation method since it is simple to customize, open source, user-friendly, platform agnostic, adaptable, saves a tone of time, and aids developers in consistently delivering Java-based web applications and building and testing the programming.

Ansible is a potent Information systems automated test engine used by countless businesses to optimize their environments and speed up DevOps approaches. Due to its flexibility, you can orchestrate the whole application environment regardless of where it is deployed[4]. The Ansible tool was developed in Python. Using the declarative markup language YAML, it specifies the devices' and settings' intended states. With the help of the comprehensive, integrated Ansible automation platform, productivity may be increased. We can automate several IT tasks with Ansible A few of these complexities is allowing for less outages for continuous integration and continuous deployment (CI/CD). To achieve this, it has frequently been necessary to use a large amount of in-house generated glue, various software packages, and bespoke code. This provides every one of these features in a compact design because it was built from the bottom up to precisely facilitate these sorts of scenarios.

Fig. 3. Ansible Architecture



We used Ansible during the CD process and adhered to DevOps best practices. As a result, we were able to transmit commands straight to the Kubernetes cluster. The application's source code is instantly recognized and, in only 37.6 seconds, sets off a chain of events involving six distinct technologies. Our knowledge of CI/CD technologies led to the creation of this article. Most businesses are thinking about switching from conventional IT methods to DevOps, but the transition will be expensive and time-consuming. There isn't much opportunity for experimenting with novel designs and deployment strategies because organization's primary objective is to offer a stable and dependable product. However, we chose to test out some creative software tool combinations for both CI and CD.

III. LITERATURE REVIEW

The research undertaken in the area of GitHub actions which allows us to automate build, test, and deployment process. The goal is to design a CI/CD pipeline with GitHub Actions to automate the development workflow. GitHub Actions removes the hassle of manually running commands on PC and pushes them to the cloud to automated runners. GitHub Actions extends beyond DevOps by allowing to perform process when additional events occur in repository The GitHub Actions process will be triggered whenever an event occurs in the repository, such as the initiation of a pull request or the creation of an issue. The CI/CD workflow with GitHub Actions is the major topic of this article has been and continues to be, the realm of DevOps professionals. With the addition of native CI/CD to GitHub in 2019 via GitHub Actions, it is easier to include CI/CD straight into workflow from repository. CI and CD are two acronyms that are often used in current development methods and

DevOps. Continuous integration (CI) is a key DevOps best practices in which developers often integrate code changes into a single repository where automated builds and testing run. But CD can refer to either continuous delivery or continuous deployment. A container orchestration platform will be required to automate the deployment. Kubernetes is a popular container orchestration technology. Because of Kubernetes resilience, users may install, scale, and manage containerized applications. AWS Code Pipeline is a critical feature that simplifies the management of CI/CD toolset. It connects with platforms like Jenkins, GitHub and Code Deploy allowing to oversee the flow of app updates from development to production graphically. AWS CloudFormation, AWS EC2, AWS CloudWatch and AWS Code Pipeline are some of the reasons which make AWS DevOps highly popular.

CI/CD is used to frequently deliver applications by incorporating automation into application development stages – build, test, merge, release and deploy. It is used to solve the issues that can be caused by integrating the new code. This paper employs Jenkins for Continuous integration and Ansible, Kubernetes for Continuous Deployment. Continuous delivery is made possible via Kubernetes. Jenkins is a Java-based CI automation platform that supports plugins that let users integrate systems like Git, Amazon EC2, and K8s.

IV. IMPLEMENTATION

Enabling Amazon EKS to create dynamic Jenkins Agent(s) to carry out application and infrastructure deployment, we may hence execute Jenkins workloads. [5] We will use the eksctl basic CLI tool for establishing clusters on EKS to configure our Amazon EKS cluster with Jenkins. The Jenkins Manager and Jenkins Agent images will then be created. To obtain and utilize the Jenkins software, the interactive Jenkins Operator pods to carry out tasks, we will then launch a container deployment on our cluster.

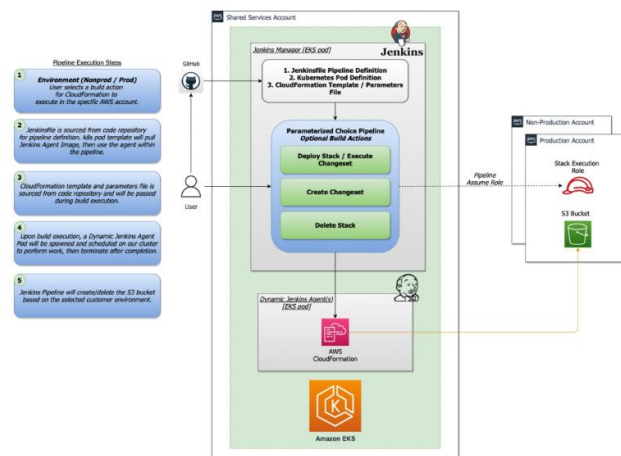


Fig. 4 Solution Architecture

In a Shared AWS account's EC2 compute instance, a Jenkins manager is executing as a container. This Jenkins application depicts distinct deployment pipelines for one-of-a-kind microservices that are built and deployed to various environments across several AWS accounts. The target AWS account admin credentials are used in the cross-account deployment to carry out the deployment. Sharing the account credentials with outside parties is not a good idea, according to this methodology. Additionally, the risk of deployment failures should be minimized, and application isolation should be maintained inside the same account. We will concentrate on using AWS CLIs in our solution because the deployment steps are carried out using AWS CLIs.

The danger is considerably reduced when utilizing CloudFormation / CDK to perform deployments since the AWS CLIs performed from the build jobs will supply stack names as parametrized inputs

and because there is very little probability that a stack-name mistake will occur. Still not recommended: using the target account's admin credentials. To complete this deployment procedure, we will set up a Profile for Centrally controlled Hosted Services for the EKS Group from Amazon. Containerized images are saved in the Jenkins Manager and Agent, create the Amazon ECR Repository. Deploy the Jenkins Manager's Kubernetes manifest file.

Create an IAM roles for the ability to cross-access the Target account from the Share services account .Jenkins plugins: Install and set up the Cloud Bees plugin and the Kubernetes. By Using a Jenkins parameterized pipeline, fetch the Jenkins file.

NAME	READY	STATUS	RESTARTS	AGE
jenkins-manager-85db4456-2vxp	1/1	Running	0	26h
test-pipeline-65-j7672-k8h79-vxd	0/2	Pending	0	8s
test-pipeline-65-j7672-k8h79-vxd	0/2	Pending	0	8s
test-pipeline-65-j7672-k8h79-vxd	0/2	ContainerCreating	0	8s
test-pipeline-65-j7672-k8h79-vxd	2/2	Running	0	2s
test-pipeline-65-j7672-k8h79-vxd	2/2	Terminating	0	48s

Fig. 5. Execution Result

This article assisted you in establishing infrastructure based on Amazon EKS and incorporating Jenkins to manage workloads. We illustrated how to use this to safely implement across different profiles using flexible Jenkins agent.

V. CONCLUSION AND FUTURE WORK

An automated CI/CD pipeline is shown in this article for launching a Java-based web application on AWS. We used Ansible during the CD process and adhered to DevOps best practice. This improved overall scalability and allowed us to transmit commands directly to the Kubernetes cluster. We achieved good results, demonstrating the dependability, scalability, and speed of the suggested approach. Therefore, each modification to the application's source code is instantly recognized and, in only 37.6 seconds, starts a series of actions involving six distinct technologies.

The effectiveness of the suggested approach, which functions flawlessly even on the least powerful computers provided by the Cloud, was underlined using CloudWatch to track resource use in AWS. The methods and procedures outlined in this article are an excellent place to start when hosting the fundamental components of the 5G network in Docker containers under Kubernetes management. In turn, this permits the development of network slices that aid in ensuring the optimum service for various applications, including augmented reality, smart cities, and autonomous vehicles.

REFERENCES

- [1] Sai Priya Sinde, Bhavika Thakkalapally, Meghamala Ramidi, Sowmya Veeramalla "Continuous Integration and Deployment Automation in AWS Cloud Infrastructure" International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538 Volume 10 Issue VI June 202
- [2] Artur Cepuc, Robert Botez, Ovidiu Craciun, Iustin-Alexandru Ivanciu, Virgil Dobrota "Implementation of a Continuous Integration and Deployment Pipeline for Containerized Applications in Amazon Web Services Using Jenkins, Ansible and Kubernetes". 2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet) | 978-0-7381-1265-7/20/\$31.00 ©2020 IEEE | DOI: 10.1109/ROEDUNET51892.2020.9324857

- [3] Sriniketan Mysari,Vaibhav Bejgam."Continuous Integration And Continuous Deployment Pipeline Automation Using Jenkins Ansible "2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE) 978-1-7281-4142-8/20/\$31.00 ©2020 IEEE 10.1109/ic-ETITE47903.2020.239
- [4] “The NIST Definition of Cloud Computing”, Computer Security Resource Center, NIST 2020
- [5] The Kubernetes Authors. Kubernetes API Server—Security. 2020.
- [6] The Kubernetes Authors. Kubernetes Official Website. 2020.
- [7] Diouf, G.M.; Elbiaze, H.; Jaafar, W. On Byzantine Fault Tolerance in Multi-Master Kubernetes Clusters. arXiv 2019, arXiv:1904.06206. [Google Scholar]
- [8] Vayghan, L.A.; Saied, M.A.; Toeroe, M.; Khendek, F. Kubernetes as an Availability Manager for Microservice Applications. arXiv 2019, arXiv:1901.04946. [Google Scholar]
- [9] Arundel, J.; Domingus, J. Cloud Native DevOps with Kubernetes: Building, Deploying, and Scaling Modern Applications in the Cloud; O’Reilly Media: Newton, MA, USA, 2019; ISBN 978-1492040767. [Google Scholar]
- [10] S. Watts, M. Raza, “SaaS vs PaaS vs IaaS: What’s The Difference & How to Choose”, BMC Software, 2019
- [11] Arnaldo Pereira Ferreira; Richard Sinnott, “ A Performance Evaluation of Containers Running on Managed Kubernetes Services”, IEEE International Conference on Cloud Computing Technology and Science, Dec. 2019.
- [12] Aayush Agarwal, Subhash Gupta, Tanupriya Choudhury"Continuous Integration and Continuous Delivery Pipeline Automation for Agile Software Project"Management. 2018 Moratuwa Engineering Research Conference (MERCon)
- [13] "Continuous and Integrated Software Development using DevOps".2018 International Conference on Advances in Computing and Communication Engineering (ICACCE-2018) Paris, France 22-23 June 2018
- [14] Arachchi, S. A. I. B. S., Perera, I., “Continuous Integration and Continuous Delivery Pipeline Automation for Agile Software Project Management,” 2018 Moratuwa Engineering Research Conference (MERCon), doi:10.1109/mercon.2018.8421965.
- [15] Wang yiran PetroChina,Zhang tongyang PetroChina,Guo yidong PetroChina "Design and Implementation of Continuous Integration scheme Based on Jenkins and Ansible".2018 International Conference on Artificial Intelligence and Big Data.
- [16] Lianping Chen on “Continuous Delivery: Overcoming adoption challenges” , The Journal of Systems and Software 128 (2017) 72–86, Science Direct
- [17] Sagar Narendrasing Deshmukh on “A System for Application Deployment Automation”, International Conference on Innovations in Power and Advanced Computing Technologies [i-PACT2017]