Continuous Integration and Deployment Automation in AWS Cloud Infrastructure

Sai Priya Sinde[1], Bhavika Thakkalapally[2], Meghamala Ramidi[3], Sowmya Veeramalla[4]

[1,2,3]Student, [4]Project Guide, Assistant Professor, Department of Electronics and Computer Engineering, SNIST, Hyderabad

Abstract-The primary objective of this project is to figure out how an application is automatically deployed. To automate the deployment, we'll require a container orchestration platform. Kubernetes is now a widely used container orchestration tool. Users can deploy, scale, and manage containerized applications thanks to Kubernetes' robustness. Continuous Integration and Continuous Delivery (CI/CD) of a project is required in modern development, and the focus is on developing and executing tests on every commit to ensure your test environment is constantly up-to-date. From the integration and testing phases through the delivery and deployment phases, CI/CD brings automation and continuous monitoring of programmers throughout their lifecycle. GitHub Actions is one of the CI/CD technology available in the market. By using GitHub as a source code management tool, it's best to use GitHub Actions as a continuous integration and delivery solution because it's supplied by GitHub. This does not require any setting or setup. As a result, incorporating these tools will speed up the deployment of an application.

Keywords- GitHub Actions, Amazon Web Services, Continuous Integration and Continuous Deployment, Kubernetes, Workflows.

I. INTRODUCTION

Every company wants to shorten the time to deliver their product into the market in order to improve employee customer service. As a result, we took the DevOps approach and used appropriate technologies to launch an application. DevOps can also lower the cost of the server environment (Amazon Web Services (AWS)) by reducing server loads, which improves server response time and automates actions. For example, Adding resources to a virtual computer. DevOps enables the company to go to market faster.

Our approach's major goal is to automate the development and deployment process so that the organization's productivity increases, and there are less failures and faster deployments. Deployment automation is a software deployment method that enables companies to release new features more quickly and often. Manual deployment is the unsophisticated technique. Human errors are possible with this procedure. Because people can't copy things as precisely as automated systems can, there's a risk for a lot of mistakes. Errors in the pre-production processes lead to errors in the production, and the manual process has a higher rate of these errors. So, for automated application deployment, we're utilizing GitHub actions as our CI/CD platform, and we're designing a workflow that builds and deploys code using the AWS EKS cluster.

II. LITERATURE SURVEY

The motivation behind this exploration was to research the way in which programmers expect to use GitHub actions to robotize work processes and the effect of activity reception on pull demands. They accumulated and broke down information from 3,190 dynamic GitHub repositories and found that small set of these repositories utilized GitHub actions, as well as 708 extraordinary predefined actions being utilized in work processes. They additionally assembled and dissected GitHub actions related issues, finding that most of the remarks were positive. By and large, the discoveries show that GitHub's actions were generally welcomed by designers[1].

This article examines the many methodologies and innovations that can be utilized to fabricate a fruitful CI/CD pipeline. They dealt with various apparatuses as well as the issues they confronted while carrying out ceaseless improvement processes in programming advancement using Devop's[2]. This paper presents the thought of on-request benefits, which alludes to the utilization of cloud assets on request and the capacity to scale assets as per request, as well as a survey of safety research in the field of cloud security [3].

This paper furnishes a mechanized work process that beginnings with distinguishing changes in the source code of a java-based web application, produces new assets in the Kubernetes group to have this new rendition, and afterward sends the containerized application to AWS. The outcomes uncovered that the proposed arrangement is trustworthy, with no margin time, is promptly versatile, and quick, and that assuming there is a shortcoming in the gig while conveying another rendition of the program, the framework returns to the latest secure form[4].

The exhibition of holders running in facilitated administrations like Microsoft Azure Kubernetes Services (AKS), Amazon Elastic Container Service for Kubernetes (EKS), and Google Kubernetes Engine is depicted in this article[5]. By conducting a

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 2022- Available at www.ijraset.com

huge scope experimental examination on GitHub projects, this paper helps experts in acquiring profound bits of knowledge about the GitHub Actions[6].

III.   OVERVIEW OF DEVOPS TOOLS

GitHub is a site that has Git-based programming improvement and form control frameworks. It has its own arrangement of elements, as well as Git's circulated rendition control and Source Code Management (SCM) abilities. GitHub Actions is a CI/CD pipeline that computerizes the build, test, and deploying pipeline. We can develop schedules that form and test each pull demand prior to conveying them to production[8]. GitHub Actions reaches out past DevOps by permitting you to lead work processes in light of different events in your repository. For instance, whenever somebody records another issue in your repository, you might execute a work process to automatically add the appropriate labels.

The GitHub Container Registry, another GitHub service for distributing and keeping up with Docker images inside GitHub, is currently available to everyone. The GitHub Container Registry allows hosting and managing Docker container images in a personal or organisation account on GitHub.

AWS (Amazon Web Services) is Amazon's far reaching, consistently developing distributed computing stage, which contains Infrastructure as a Service (IaaS), Software as a Service (SaaS), and Platform as a Service (PaaS) offerings. AWS services offer organisation tools such as compute power, content delivery services and database storage.

The Amazon Elastic Kubernetes Service (Amazon EKS) is an overseen Kubernetes administration that makes running Kubernetes on AWS and on-premises straightforward. Kubernetes is an orchestration tool which is used for automating deployment, scaling, and management of containerized applications. Since Amazon Elastic Kubernetes Service is affirmed Kubernetes-agreeable, current Kubernetes-based applications will work with Amazon EKS[12].

The AWS Command Line Interface (CLI) is for managing AWS services from a terminal session on our own client, which allows you to control and construct multiple AWS services and implement a level of automation.

Kubernetes provides a client side tool for interfacing with a Kubernetes cluster control plane. KUBECTL is the name of this tool.

IV.   PROPOSED SYSTEM

GitHub Actions is an event-driven continuous integration and continuous delivery tool which allows us to automate the process of building, testing, and deploying the application into various domain. By event driven, we imply that a GitHub actions work process runs because of an event that happens inside the GitHub actions, for example, a pull request and a push to a branch, and so on. GitHub Actions is easy to use. All we need to get started is a YAML file. If a developer has some knowledge of YAML so GitHub Actions is a logical choice as a CI/CD platform. While coming to setup and maintenance, **w**ith Jenkins, it will be running on a custom server. This means that we have to maintain the Jenkins server continuously. However, GitHub Actions provides free runners which we can use them to perform CI/CD operations. These runners are owned and maintained by GitHub but we can add self-hosted runners as well. So, we are using GitHub Actions as a Continuous Integration and Continuous Deployment tool in our project for automation purpose. The Amazon Elastic Kubernetes Service (Amazon EKS) is an overseen Kubernetes administration that makes running Kubernetes on AWS and on-premises straightforward. So by integrating GitHub Actions with AWS EKS cluster we are deploying the application.

Fig. 1. The Proposed Architecture

V.   IMPLEMENTATION

A.   Creation of Web Application

To make a website, we require an IDE(Integrated Development Environment) to foster the code. Visual studio code, Komodo Edit, Notepad++ are a portion of the free, open source IDEs that you can be utilized for HTML. Here we are utilizing Visual

studio code. Since it offers fundamental help for HTML programming.

B. Workflow Automation

In request to run this web application, we are involving GitHub Actions as a Continuous Integration and Continuous Deployment device for mechanizing the sending. GitHub Actions utilizes YAML language structure to develop the work flow. Every work flow is put away as a different YAML record in code store, in a catalog named .github/workflows[20].

Create a workflow. To manually trigger the workflow we use Workflow dispatch action. Manual triggering is done when there are no changes in the code. To run the workflow, we need an operating system or virtual machine. GitHub Actions provide hosted virtual machines like Ubuntu, MacOS, Windows. We have taken Ubuntu-latest as a base server so that the jobs will be executed on this fresh virtual machine.

To connect to the cluster, provide credentials such as *access_key_id, secret_access_key, session_to*ken of AWS EKS cluster. For security reasons these credentials are kept in secret folder of our repository. And provide the region in which our AWS cluster is working.

This workflow uses docker/login-action to authenticate to registry. The Container registry will support the GITHUB_TOKEN for secure and easy authentication in the workflows. But now it is necessary to create a Personal Access Token with correct scopes and add it as a repository secret. You can get the PAT from developer settings of GitHub's personal settings. Once you get the token copy it, navigate to repository settings and create a secret called GHCR_PASSWORD and insert the token as the value. GitHub Actor is the username of the user that initiated the workflow run. This workflow uses docker/build-push-action to build the image and push it to the registry. You can trigger it by creating a Git tag that matches the version format.

Once you run the workflow, you can see the image will appear under the repository owner's profile within the Packages tab. From your profile, you can manage versions and link it to a repository, or can change the visibility from private to public. If the image is private, you'll need to authenticate with the GitHub Container Registry wherever you intend to pull the image as well. Here, we are making the image as public to have external access.

GitHub Actions uses kubectl to manage kubernetes cluster. To install a specific version of kubectl binary on runner, use this action in the workflow to define which version of kubectl will be used. Once the image is published to the registry, we need to run this image on AWS EKS cluster to deploy the web application. We are using a resource named deployment to run this application. This deployment is created using a yaml file. To have external access to the application, expose the deployment using a resource called service.

To externally access this application we need to have an IP address which you will get by running get service command in the cluster. But if any developer wants to make some changes or add new features to this web application, then when developer commits the changes in the code, the docker image is created automatically with the new features and it will be updated in the container registry. Now we need to use this updated image for deployment using Kubernetes cluster. Kubernetes provides a feature to rollout to the newer version of the image, we can set the image for a container using command line with the help of container name. An action will update the image and clients can access the updated application.

While running the application with the new features, if the build fails suddenly due to the presence of errors in the code, at this time there is downtime for the clients in accessing the application, the clients can access the previous application at these times.

VI. RESULTS

After building and running the workflow successfully, the application is deployed on the Kubernetes cluster. We can access the application using external IP address provided by Load Balancer. The application is accessed by the clients all over the world with this IP address when they are connected to the internet.

The external IP Address of our web Application is "http://ab44a6bc4dc4c4d2a9cd09b6b66c848d-325833986.us-east-1.elb.amazonaws.com/." This is the output of our web application
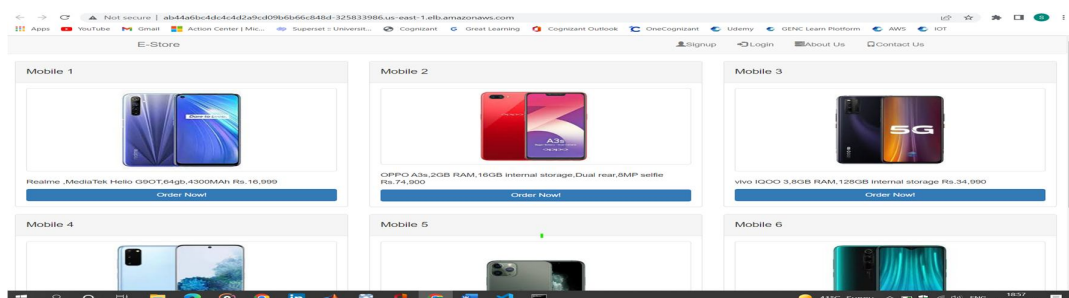

Fig. 2. Web Application

VII. CONCLUSION

The approach we used to deploy an application is by using GitHub actions which minimise the number of tools used. Using GitHub actions as a CI/CD pipeline with the AWS EKS cluster, the Flow of deployment is continuous and errorless. The time taken for the deployment of an application with Jenkins as a CI/CD pipeline is 1 min 43 sec approximately. But the time taken for our approach to deploy an application is around 25 sec to 45 sec.

REFERENCES

[1] Timothy Kinsman, Mairieli Wessel, Marco A. Gerosa and Christoph Treude , "How Do Software Developers Use GitHub Actions to Automate Their Workflows?", IEEE 18th International Conference on Mining Software Repositories, 2021.

[2] Artur Cepuc, Robert Botez, Ovidiu Craciun, Iustin-Alexandru Ivanciu and Virgil Dobrota, "Implementation of a Continuous Integration and Deployment Pipeline for Containerized Applications in Amazon Web Services Using Jenkins, Ansible and Kubernetes", 19th RoEduNet Conference: Networking in Education and Research, 2020.

[3] Aayush Agarwal, Subhash Gupt and Tanupriya Choudhury, "Continuous and Integrated Software Development using DevOps", International Conference on Advances in Computing and Communication Engineering, June. 2018.

[4] Saakshi Narula, Arushi Jain and Ms. Prachi, "Cloud Computing Security: Amazon Web Service", Fifth International Conference on Advanced Computing & Communication Technologies, 2015.

[5] 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.

[6] Tingting Chen, Yang Zhang, Shu Chen, Tao Wang and Yiwen Wu, "Let's Supercharge the Workflows: An Empirical Study of GitHub Actions", IEEE 21st International Conference on Software Quality, Reliability and Security Companion, Dec. 2021.

[7] GitHub," https://en.wikipedia.org/wiki/GitHub"

[8] GitHub Actions," https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions"

[9] GitHub Marketplace," https://docs.github.com/en/developers/github-marketplace/github-marketplace-overview/about-github-marketplace"

[10] GitHub Container Registry," https://www.infoworld.com/article/3623291/github-container-registry-available-for-production-use.html"

[11] Amazon Web Services," https://www.techtarget.com/searchaws/definition/Amazon-Web-Services"

[12] Amazon Elastic Kubernetes Service," https://aws.amazon.com/eks/features/"

[13] AWS Command Line Interface," https://cloudacademy.com/blog/how-to-use-aws-cli/"

[14] KUBECTL,"https://kubernetes.io/docs/reference/kubectl/#:~:text=Kubernetes%20provides%20a%20command%20line,config%20in%20the%20%24HOME%2F."

[15] Docker," https://www.ibm.com/in-en/cloud/learn/docker"

[16] HTML," https://en.wikipedia.org/wiki/HTML"

[17] CSS," https://en.wikipedia.org/wiki/CSS"

[18] PHP," https://en.wikipedia.org/wiki/PHP"

[19] YAML,"https://www.redhat.com/en/topics/automation/what-is aml#:~:text=YAML%20is%20a%20data%20serialization,is%20for%20data%2C%20not%20documents."

[20] Workflow," https://docs.github.com/en/actions/using-workflows/about-workflow

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 ⓦ (24*7 Support on Whatsapp)