Best Practices for CI/CD Pipelines Implementation in Cloud Native Applications

Abstract —The report focuses on illustrating the advantages and challenges of CI/CD pipelines implementation in cloud native applications in order to highlight the great value of CI/CD pipelines in cloud-native application development. Besides, the report provides the best practices and the case in real world about CI/CD pipeline implementation in cloud native applications, aiming to demonstrate how to maximize the value of CI/CD pipeline in cloud native application development.

I. INTRODUCTION

Nowadays, due to the rapid development of cloud platforms, the demand for cloud-native applications is tremendously increasing. How to quickly develop cloud-native applications while ensuring security and reliability during the development process has become a problem that requires much attention. The CI/CD pipeline can effectively solve this problem under the cloud native framework. This report illustrates the enormous value of CI/CD pipeline implementation in terms of advantages, challenges, best practices, and cases.

II. THE ADVANTAGES OF CI/CD PIPELINES IMPLEMENTATION

Speaking of the advantages of CI/CD pipelines implementation in cloud-native application development, first, the CI/CD pipeline reduces the need for human software construction and testing, which not only increases the development speed of cloud native applications, but also reduces the possibility of errors and improves the reliability of the code. Secondly, due to the advantage of detecting and debugging errors faster, it allows programmers to focus more on the development of cloud native programs, and therefore the speed of software upgrades and iterations is increased. Because of the improvement of development efficiency, cloud-native applications' release cycles are accelerated, and development costs are effectively reduced. Moreover, the CI/CD pipeline can improve the transparency in the development process because the CI/CD pipeline can continuously help code integration, which helps the code synchronization of all people in the team and make the development process within the team more transparent. In addition, because the development team can continuously receive feedback from the CI/CD pipeline, it allows the development team to have a clearer understanding of the development of cloud native programs, which also increases the transparency.

III. THE CHALLENGES OF CI/CD PIPELINES IMPLEMENTAION

For the challenges of CI/CD pipelines implementation, first, because CI/CD pipelines are generally provided by third parties, if the services provided by the third-party encounter network attacks, there may be a risk of leakage of the codes and data of cloud-native application. Because microservices under cloud-native frameworks may have complex dependencies, handling dependencies has become a challenge. Programmers need to clarify the structure of the microservices and provide an isolated environment for them to ensure that different microservices operate normally under dependency relationships. In addition, if a company wants to deploy large-scale cloud-native application on the CI/CD pipeline, it may involve many kinds of services, processes and complex environments. Therefore, there are great

challenges in the deployment due to this reason. Moreover, because the process of detecting and optimizing software in the CI/CD pipeline is often based on functions that have not been fully implemented, providing high-quality optimization and detection for functions that have not yet been fully completed is a challenge.

IV. COMMON TOOLS AND TECHNOLOGIES

Take GitHub Actions, Tekton, and Jenkins as examples to introduce these common CI/CD pipeline implementation tools. First, Github Actions. Github Actions is a platform implemented by Azure that can achieve continuous integration and continuous delivery. It can test cloud native applications under development through automated construction and check whether there are errors. Second, Tekton. Tekton is a cloud-native framework based on Kubernetes and Custom Resource Definitions. It has the function of testing and deploying cloud-native applications across multi cloud platforms and local systems to achieve good continuous integration and continuous delivery. Tekton is used by many large-scale companies such as Google and IBM. Third, Jenkins. Jenkins is a Kubernetes based continuous integration and continuous deployment platform. Jenkins integrates many tools to build an automatic compilation environment, test, and deploy applications and generate container images.

V. BEST PRACTICES IN CLOUD NATIVE APPLICATION

First, speaking of automatic tests and deployment, the best practice is that after decomposing the functions to be tested, programmers need to isolate them into different environments for testing and frequently do testing. The reason is to ensure accurate measurement of the specific situation of the function while avoiding potential interference.

Second, speaking of incorporating security and compliance, the best practice is that although CI/CD pipeline implementation is mostly automated, programmers need to regularly monitor its security. Besides, in the CI/CD pipeline, it is necessary to set protection for the code and data to ensure the security of the cloud native application during the development process.

VI. SUCCESSFUL CI/CD PIPELINE IMPLENTATION CASE IN CLOUD-NATIVE APPLICATION

In July 2020, Agricultural Bank of China (ABC) released a lightweight cloud-native application platform and realize the CI/CD pipeline implementation successfully in it. By using Tekton as a cloud native framework, a series of steps in application development including building, testing, deployment have been achieved. At the same time, by using the containerization function in Tekton for application environment construction, the efficiency of software development can be improved. Because CI/CD pipeline is implemented on this cloud-native application platform, Agricultural Bank of China has won a great advantage in the market.

VII. CONCLUSION

Thanks to the advantages of CI/CD pipelines that it can improve the security and reliability of cloud native applications during development while saving programmers time in manual detection for cloud native applications, implementing CI/CD pipelines has become one of the essential tasks in the process of cloud native application development. The use of CI/CD pipelines during the process of cloud native application development is highly encouraged.