International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

Analysis of CI/CD Application Based on Cloud Computing Services on Fintech Company

Asfin Achdian[1], M Akbar Marwan[2]
[1]Master of Technology and Engineering, Gunadarma University, Depok, Indonesia
[2]Master of Technology and Engineering, GunadaramaUniversity, Depok, Indonesia

Abstract**—** *This thesis aims to analyze the application of integration, deployment, and delivery (CI/CD) using cloud service on fintech company by using qualitative content analysis methods. In the process of making a system or application that development, testing, bug-fixing, and implementation in production environment. In these stages there are three align the functions of the three elements. Therefore, this thesis will discuss the application of the CI/CD concept using cloud computing services that are implemented in fintech company using AWS (Amazon Web Services) as a VPS (Virtual Private Server), and GitLab as the CI/CD application.*

Keywords**—** *Continuous Integration, delivery and development (CI/CD), cloud computing, Fintech.*

I. Introduction

The creation of a continuous system or application in the information technology life cycle includes several elements, namely planning, design, development, testing, error correction, continued testing, and implementation in a production environment. In these stages there are three divisions involved, between developers, testers and operational units. For developers to concentrate on the speed of making the program code and the functional amount implemented into the program, the examiner will see how many errors and bugs are detected, for the operational unit will see in terms of system stability and the small risk of a system failure.[1]

With such workflows often leading to conflicts of interest, on the developer side trying to write program code quickly and sending it to testers, testers examine and test program code as long as necessary to reveal all bugs, and third parties operating departments are hesitant to make major changes. in the program code, because this change has the potential to open access to all IT infrastructure to serious risks.. On the other hand business teams must be flexible and client-oriented, so that they can

realize new products and services in a timely manner.Therefore it is necessary to have a mechanism that harmonizes the functions of the three elements mentioned above. This thesis will analyze the application of the CI / CD concept (Continuous Integration, using cloud computing services that are implemented in financial technology companies using AWS as a VPS (Virtual Private Server), using GitLab as a CI / CD application.

II. Literature Riview

The introduction of the CI concept in 1991 was one of the efforts to overcome the integration problem during the project. Continuous integration is one of the 12 extreme initial programming experiments [2]. In some cases, the practice of its use in an organization can vary but in broad outline CI still consist of:

a. The commit code changes to the main repository at least once a day by each contributor.
b. These changes will usually be first verified by each team member in the local developer environment.
c. When pushed to the code repository, new changes will be verified automatically and will build with tests.
d. If there is a change that causes a build or a failed test, then this must be corrected immediately.

The thing that often happens, if you build a new code that will be successfully deployed to the staging environment which is a replication of production. [3]. Automatic deployment practices are under two things: continuous delivery and continuous deployment. Both of these will be referred to as CDs and CDs. First of all, what are the definitions of CDs and CDs and what are the differences? According to Caum's blog posts for playing Puppet automation tools, Sustainable Delivery is a series of practices that ensure the fast, constant and reliable delivery of the latest change codes for production environment. [4]

III. Analysis and Solution Design

From the results of observations and observations made on the system that is running on the company, there are several points, namely the need for a workflow system that can support the needs of the team.

For this reason the author provides a solution to answer the needs of the observations and observations that have been done, namely by applying CI / CD which is the process of implementing DevOps using GitLab, GitLab was chosen because GitLab is the first tool or application that covers the entire DevOps cycle from the planning stage to the monitoring phase, thus providing efficiency and accelerating business continuity and Helps IT teams to manage, plan, align and track each project to ensure the team is working properly, in the right time and maintain program accuracy throughout the shipping cycle from idea to production. [5]

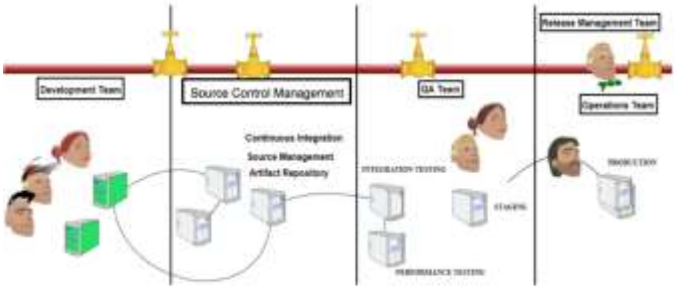


Fig. 1. CI/CD Pipeline

In the initial stages of CI / CD the Pipeline is the developer environment and CI environment, then enter the next stage of Source Control Management, then QA Testing, and the final stage is the release to Production Environment.

IV. Implementation

At this stage the authors implement the results of the analysis and design of the DevOps pipeline into the CI / CD system form, here are some things that must be prepared.

Cloud Server

This Cloud Server is prepared to install GitLab, as for the Specifications as follows. [6] [7]
- 64-bit Amazon Machine Image Ubuntu Server 18.04 LTS
- Instance Type t2.large
- vCPUs 2
- 8 GiB memory
- SSD Volume size 20 GiB type IOPS 100 gp2

Gitlab

In the installation process used is using the GitLab Omnibus feature, this feature is widely used because it is very easy in the installation process which includes GitLab configuration settings, GitLab database, HTTPS, Email Notification, elasticsearch, and the registry container. [8]

Role User

In addition to the administrator user who plays a role in setting up GitLab Server, it is necessary to add users according to the CI / CD analysis and design that will be implemented, namely the business user team, developer, tester, and release manager.

Branching Strategy

Branching is a very useful feature provided by GitLab, this feature is very helpful for developers in working on and developing several features in parallel, Branching can also define environments such as development, test, staging, and production. In this writing the program code in the development phase will be placed in the branch development, then the program code at the testing stage will be placed in the branch testing, and so on until the production program code is placed in the master branch, where GitLab has set the master as default when first making Git Repository. [9]

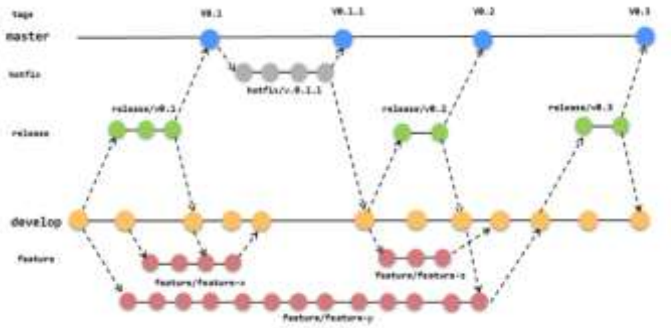
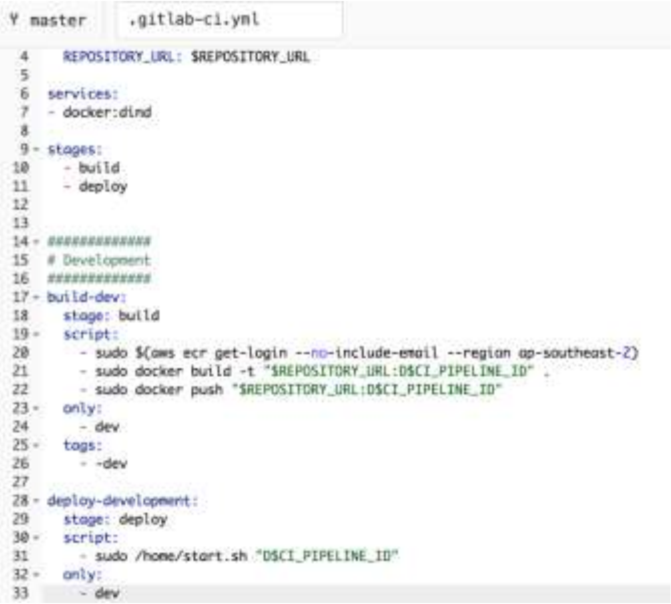


Fig. 2. Branching Strategy

Configuration file gitlab-ci.yml

The gitlab-ci.yml file is a CI / CD script that will be run by the gitlab-runner service, the script is a command to run the build command shell docker then the image will be saved into the repository, the image in the repository will be installed to environment development, staging and production. The service will be triggered with the merge request command that is run on GitLab [10] [11].



```
  REPOSITORY_URL: $REPOSITORY_URL

services:
- docker:dind

stages:
  - build
  - deploy


#############
# Development
#############
build-dev:
  stage: build
  script:
    - sudo $(aws ecr get-login --no-include-email --region ap-southeast-2)
    - sudo docker build -t "$REPOSITORY_URL:D$CI_PIPELINE_ID" .
    - sudo docker push "$REPOSITORY_URL:D$CI_PIPELINE_ID"
  only:
    - dev
  tags:
    - -dev

deploy-development:
  stage: deploy
  script:
    - sudo /home/start.sh "D$CI_PIPELINE_ID"
  only:
    - dev
```

Fig. 3. Script gitlab-ci-yaml

V. Conclusion

Based on the formulation of the problem and the series of research that the author has done. Then it can be concluded several things as follows.

a. The way to facilitate, speed up and synchronize the information system development workflow with the *DevOp* method in this case in *fintech* companies is to implement CI/CD using *GitLab* based on cloud computing with a pipeline that starts from planning and ideas to the production environment.
b. The related work section includes team product, team development, tester, then release manager.
c. By applying CI/CD it is useful to monitor a project in terms of speed of making program code to deployment to

production environment.

d. The application of CI/CD functions as an automation of the performance of all teams related to a development project.

ACKNOWLEDGEMENT

The authors would like to thank Research Center in the College of Computer and Information Science, Gunadarma University, and Software Development Team Mekar.id for its support to complete this research.

REFERENCES

[1] R Ganesan V. (2017). Blameless Continuous Integration: A Small Step Towards Psychological Safety of Agile Teams. p 5.

[2] Fowler M. (2018). Continuous Integration [Online]. Available at https://martinfowler.com/articles/continuousIntegration.html [Accessed 27 January 2018].

[3] Vadapalli S : Dive into the core DevOps Strategies, 2018

[4] Grubor S : Apply continuous integration models, deployment application quicker, and scale at large by putting Docker to work, 2017

[5] Jeroen V.B : GitLab Cookbook, 2013

[6] van Bennekum Alistair Cockburn Ward Cunningham Martin Fowler James Grenning Jim Highsmith Andrew Hunt Ron Jeffries Jon Kern Brian Marick Robert C. Martin Steve Mellor Ken Schwaber Jeff Sutherland Dave Thomas KBMBA. Principles behind the Agile Manifesto [Internet]. 2001

[7] DevOps dictionary definition | DevOps defined [Internet]. [Access 2018 Jan 13]. http://www.yourdictionary.com/devops

[8] Yehuda Y. What's the relationship between DevOps and Continuous Delivery? [Internet]. IT Pro Portal. 2015 [Access 2018 Jan 14]. https://www.itproportal.com/2015/06/09/whats-relationship-between-devops-and- continuous-delivery/ 2018 State of DevOps Report [Internet]. Puppet. 2018 [Access 2018 Jan 16] https://puppet.com/resources/whitepaper/state-of-devops-report

[9] Brodkin J. With long history of virtualization behind it, IBM looks to the future [Internet]. Network World. 2009 [Access 2018 Jan 22]. https://www.networkworld.com/article/2254433/virtualization/with-long-history-of- virtualization-behind-it--ibm-looks-to-the-future.html https://www.howtogeek.com/66734/htg- explains-what-is-a-hypervisor/

[10] Bhupender. Types of Virtualization in Cloud Computing- An Overview [Internet]. ZNetLive Blog - A Guide to Domains, Web Hosting & Cloud. 2016 [Access 2018 Jan 22]. https://www.znetlive.com/blog/virtualization-in-cloud- computing/

[11] Gitlab. (2018). Gitlab Product [Online]. Available at https://docs.gitlab.com/ee/README.html [Accessed 25 Mei 2018].