

Deploying an Application to Cloud Platform Using Continuous Integration and Continuous Delivery

Muhammad Fauzan Albaihaqi^{1,*}, Anisa Nurul Wilda², Bambang Sugiantoro¹

¹Informatics Department, UIN Sunan Kalijaga

²Informatics Department, Universitas Islam Indonesia

Email*: mfauzan.albaihaqi@gmail.com

Abstract. Cloud Computing is the best way for business owner deploy an application to reduce cost issue because it is implement pay as you go concept. Generally, an application on production level or deployed into cloud instance should not have any error or bug. It should be tested and maintain properly. The problem when an application have intensive development that takes more effort to test the application and deploy. So, need a strategy to deploy an application into cloud instance to make the process more efficient. Nowadays, Version Control System (VCS) platform provide Continuous Integration and Continuous Delivery (CI/CD) feature. Users can utilize that platform to perform automated test and deployment easily. This research purposed to examine how to use CI/CD feature and evaluate it in case of deploying web application to Cloud Platform. Researcher use Gitlab which is provide CI/CD feature for free and deploy the app to Amazon Web Service. The researcher also utilize docker container to accommodate all processes. The result are Continuous Integration can improve application quality because most lines of codes are tested using unit or feature test scenario. Using CI/CD feature improve security issue of deployment. Deployment process run automatically without human intervention so it will reduce human error factors. This feature also ensure high availability of an application. Deployment process will take zero downtime. The application can quickly update without any downtime and configuration. Last, docker container take an important role for deployment of application into cloud instance.

Keywords: Deploy, Cloud Computing, Continuous Integration, Continuous Delivery, Git, Docker, Amazon Web Service

INTRODUCTION

With data, small business or large business now can rapidly growing up. For a large business, the owner usually can spend more money to build powerful and secure infrastructure to make sure their data operation. But not all business owner can spend more money to build their own infrastructure. Some business can use Cloud Computing for best solution build an infrastructure. Cloud Computing provide low cost, efficiency, and less maintainance (Yasha, 2018). Cloud is a class of software which provide some resources for development process as a service. Resources are hardware, software, environment, and some technology. The cloud services classify into Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service. IaaS is based on physical or virtual computer. IaaS provide compute resource such as server, network, storage, and data center for business. PaaS is a platform for business particularly to deploy application above it. For example Amazon Web Service Elastic Beanstalk, Google Cloud App Engine, Azure App Services, etc. Users can build, deploy, test, and configure any application and choose specific framework. This service provide a software for users which is easily to use. SaaS software for example Google Apps (Docs, Spreadsheet, Presentation), Office 365, and Adobe Creative Cloud. Users just only use the application without knowing how the data is saved or

how the application is maintained. (Charan, etc, 2011). Now, business owner can easily be deploy, develop or use application with cloud platform using Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) (Rani & Ranjan, 2014).

A development process may not have an end situation because of stakeholder requests or users requirements based on software development life cycle. Any problems will arise if development process is do by manually. As developer should not to change sourcecode directly because it is too risky if the application get any error after that changes. Application which is deployed in a cloud platform service and accessed by end user generally should not have any error or bugs. Before an application deployed in cloud platform, an application also should pass testing process. Need a strategy to execute development process and deploy an application into cloud instance to make the process more efficient and reduce all risk.

Version Control System mostly use in every software development. This platform provide strong framework collaboration. Developers can easily track the difference code if changes happen from the repository. Most importantly developers can combine their work without any conflict. VCS have high flexibility and provide strong option to manage and develop sourcecode collaboratively. (Case, 2016). Nowadays, Version Control System (VCS) platform provide Continuous Integration and Continuous Delivery (CI/CD) feature.

Users can utilize that platform to perform automated test and deployment easily (Jayachandran, n.d.).

Continuous Integration is a software development practice where all team members merge each job or sourcecode into one application repository. Then, software will automatically verified by particular testing program to test and make sure the application have no error or bug quickly. This method can reduce any problem and might be accelerte development process (Fowler, 2006). At some point Continuous Integration also can increase quality of the application. Continuous

Delivery is a software engineering approach to make sure the application can be released anytime while a developer team work together build an application. This approach intended to build, test, and release the application faster and more often. In case of any update from the current application, Continous Delivery also can reduce cost, time, and risk of changes. This research purposed to examine how to use CI/CD feature at VCS platform and evaluate it in case of deploying web application to Cloud Platform.

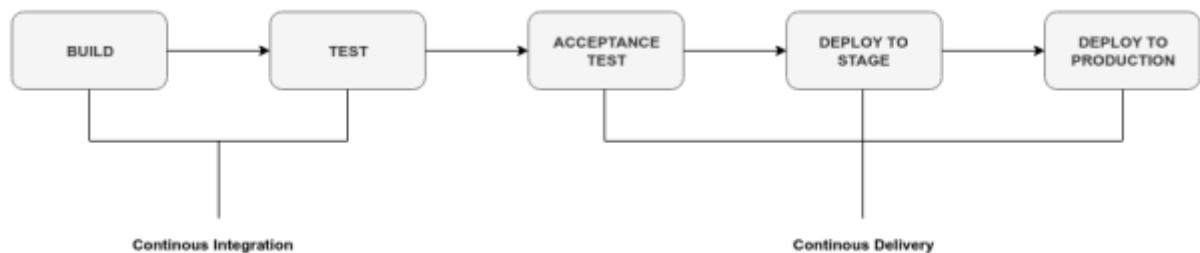


Figure 1. Continuous Integration and Continuous Delivery Process.

MATERIALS AND METHODS

This research use some platform and software to build web application, run Continuous Integration Continuous Delivery and deploy to Cloud Platform Service. Researcher use Gitlab for VCS platform wich is provide CI/CD feature for free and targeted deploy the application into Elastic Compute Cloud (EC2). Elastic Compute Cloud is one of Infrastructure as a Service Cloud provided by Amazon Web Service. It is a virtual machine or instance containing any software what users need. EC2 have public IP so it can accessed by public and possible to make it as production level. Researcher use NodeJS + Express for sample web application.

The researcher also utilize docker container to accommodate all Continuous Integration Continuous Delivery pipeline processes. Application developed using test driven development which is there are test files represented all features. Pipeline start with executing test comand over docker container. If test is failed, it means there are some error inside application so the proccess stopped. If test pipeline is passed, the application is ready to release or deploy. The sourcecode is built into docker image and push into a docker image registry. Inside cloud instance, researcher make a file containing all processes to make the application docker iamge run over instance's docker service. Last, the script executed by SSH agent from deployment stage.

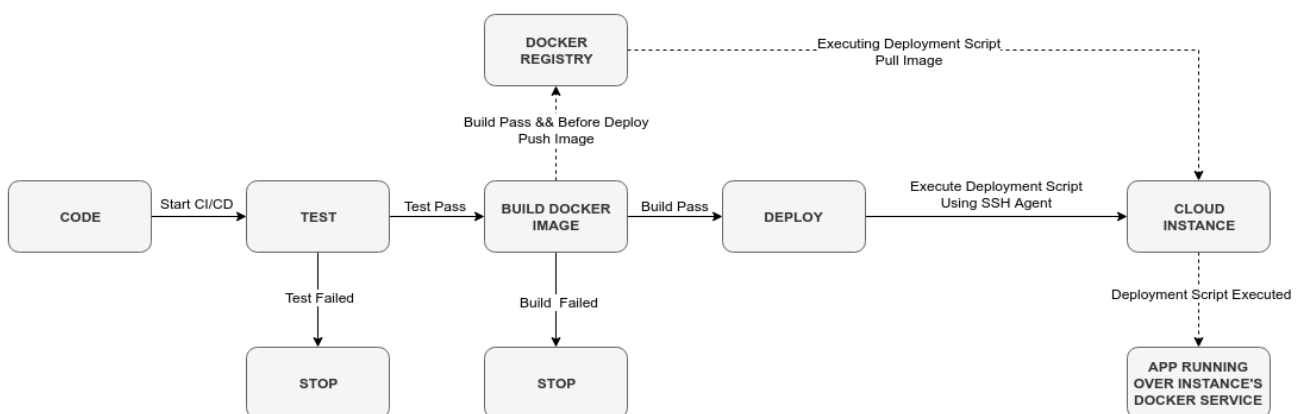


Figure 2. CI/CD Deployment Method.

RESULTS AND DISCUSSION

Gitlab CI/CD

Gitlab provide CI/CD feature basically use docker container to run each stage. While it start, Gitlab CI/CD runner load docker image and execute particular script to run stage scenario. Gitlab require a file named *.gitlab-ci.yml* inside the repository that is containing pipeline stages and comands each stages.

```
.gitlab-ci.yml 1.4 KB
1  stages:
2    - test
3    - dev-build-latest
4    - dev-deploy-server
5
6  unit-test:
7    image: docker:latest
8    services:
9      - docker:dind
10   stage: test
11   environment: test
12   script:
13     - 'docker build -t $CONTAINER_IMAGE_TEST:latest .'
14     - 'docker network create $DOCKER_TEST_NETWORK'
15     - docker run --volume "$DB_NAME_TEST:/data/db"
16       --name $DB_NAME_TEST --network $DOCKER_TEST_NETWORK
17       --hostname $DB_HOST_TEST
18       -d -p 27017:27017 mongo:4.0.6
19     - docker run --name $SERVICE_NAME_TEST
20       --network $DOCKER_TEST_NETWORK --hostname $SERVICE_HOST_TEST
21       -d $CONTAINER_IMAGE_TEST:latest'
22     - 'docker exec $SERVICE_NAME_TEST npm run test'
23   only:
24     - dev
```

Figure 3. Test Stage.

Figure 3 is a sample of executing unit test using NodeJS + Express application. This process is a part of Continous Integration. The pipeline will stop if test result return failed job. If test result is passed, sourcecode is build into docker image and push to docker hub registry at build image stage as figure 4. For the next, a computer or cloud instance can easily get or pull newest version of application image from docker hub registry.

```
26 build-docker-image-latest:
27   image: docker:latest
28   services:
29     - docker:dind
30   stage: dev-build-latest
31   environment: dev
32   script:
33     - 'ls'
34     - 'docker login -u $DOCKER_USER -p $DOCKER_PASSWORD docker.io'
35     - 'docker build -t $CONTAINER_IMAGE:latest .'
36     - 'docker push $CONTAINER_IMAGE:latest'
37   only:
38     - dev
```

Figure 4. Build Image / Acceptance Stage.

Inside cloud instance, researcher make a script containing all processes to make the application docker iamge run over instance's docker service. The process

are pull docker image newest version from docker hub, stopping current container, remove current container and run new container with new docker image version. The script executed by SSH agent from deployment stage as figure 5.

```
40 deploy-to-dev:
41   image: "ruby:2.5.1"
42   stage: dev-deploy-server
43   environment: dev
44   script:
45     - 'which ssh-agent || ( apt-get update -y && apt-get install openssh-client -y )'
46     - eval $(ssh-agent -s)
47     - ssh-add <(echo "$SSH_PRIVATE_KEY")
48     - mkdir -p ~/.ssh
49     - echo -e "Host *\n\tStrictHostKeyChecking no\n\n" > ~/.ssh/config
50     - ssh $USER_INSTANCE@$INSTANCE_IP_ADDRESS bash --login ". /deploy-dev.sh"
51   only:
52     - dev
```

Figure 5. Deployment Stage.

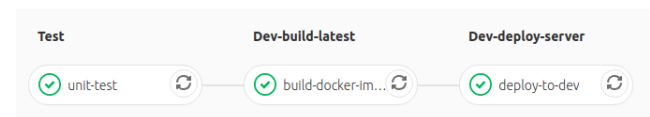


Figure 6. CI/CD Pipeline

Benefit

Using CI/CD absolutelly make development process more efficient. It work automatically so any update can easily and quickly update the production stage. Continous Integration method can make the application have a high quality of code. The code automatically integrate with testing stage. Most lines of codes are tested using unit or feature test script while stage is running. Deployment process run automatically without human intervention so it will reduce human error factors. CI/CD ensure high availability of an application. After docker service successfully pulled newest version of application image, it quickly remove application's current container and replaced with newest version. So deployment process take zero downtime that means without any downtime and configuration.

Discussion

This deployment method using Gitlab CI/CD can be possible to build and deploy to Cloud Platform Service. Docker container take an important role for deployment of application into cloud instance. There are some benefit using this deployment method. Continous Integration can improove application quality because most lines of codes are tested using unit or feature test scenario. Using CI/CD feature improove security issue of deployment. Deployment process run automatically without human intervention so it will reduce human error factors. This feature also ensure high availability of an application. Deployment process take zero downtime. The application can quickly update without any downtime and configuration.

REFERENCES

- Fowler, M., 2016. Continuous Integration, s.l.: s.n.
- Jayachandran, J., n.d. In: A. Systems, ed. Cloud Migration Methodology. s.l.:White Paper.
- Rani, D. & Ranjan, R. K., 2014. A Comparative Study of SaaS, PaaS and IaaS in Cloud Computing. International Journal of Advanced Research in Computing Science and Software Engineering, June.4(6).
- Charan, N. Ram Gangga & Rao, S. Tirupati & Srinivas, Dr. P. V. S., 2011. Deploying an Application on the Cloud. International Journal of Advanced Computer Science and Applications.
- Tsalolikhin, Aleksey, 2018. Introducing To CI/CD: Concepts and Terminology
- Case, Denise M. & Eloe, Nathan W., & Leopold, Jennifer L., 2016. Scaffolding Version Control into the Computer Science Curriculum
- Shahina, Mojtaba & Babara, Muhammad Ali & Zhu, Liming, 2017. Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices.
- Yasha, 2018, Dewaweb, Cloud Computing: Panduan Bagi Pemula. Accessed from <https://www.dewaweb.com/blog/cloud-computing>