



Practical implementation of CI/CD and platform architecture for web application development

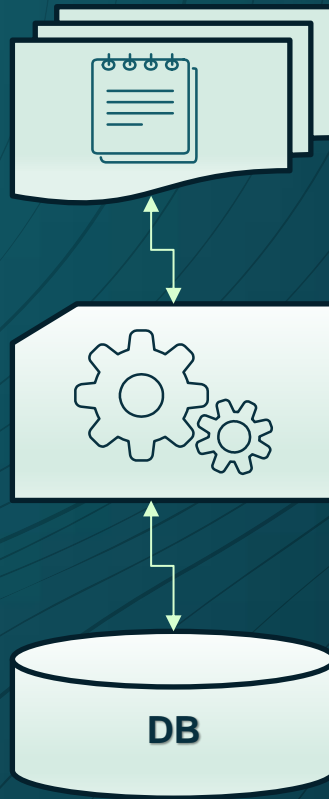
Reporter: Ihor Porokhnia
Kharkiv 2020

Abstract

- Traditional software development methodologies are not enough to fulfill nowadays business requirements
- Continuous practices, i.e., continuous integration, delivery, and deployment, are the software development industry practices that enable organizations to frequently and reliably release new features and products
- Adaptation of new practices enables flexibility, efficiency and speed of Software Development Life Cycle (SDLC), which is attracted by software development companies
- There are many different ways to organize SDLC

Application structure

- Static resources, like HTML, JS, CSS files
- Java application
- Relational database



◆ Objectives

- Increase flexibility of application platform
- Make platform quick-deployable, highly available and scalable
- Decrease gaps between SDLC steps
- Automate typical SDLC processes



Minimum changes solution

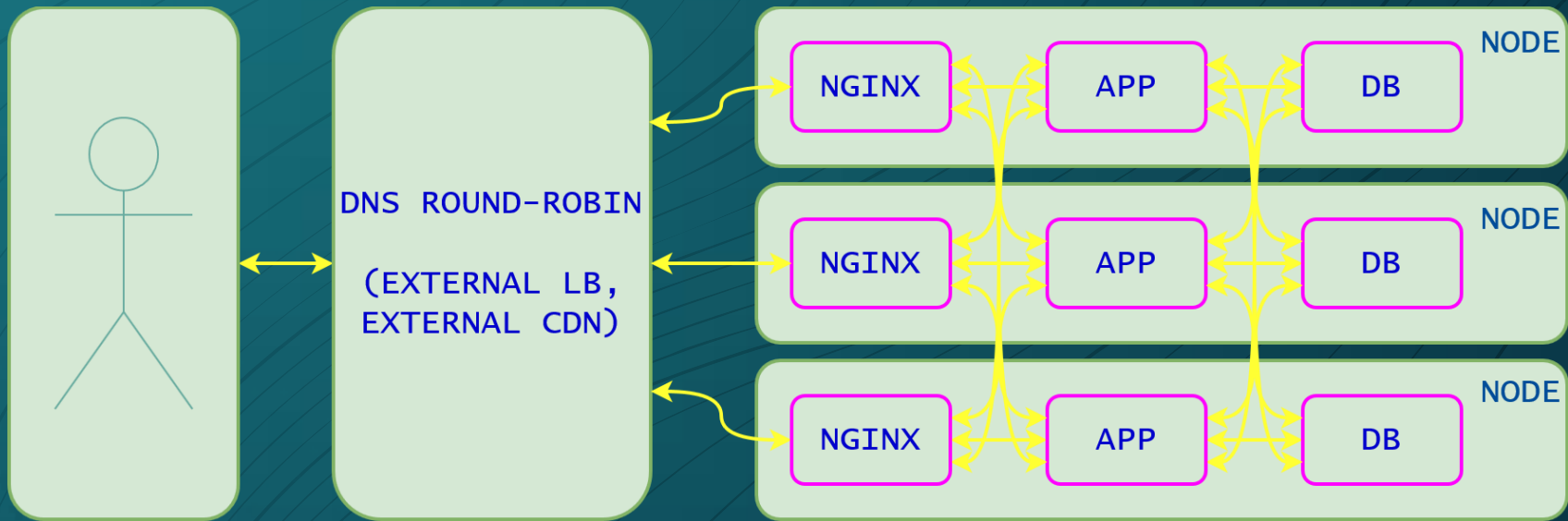
If it ain't broke, don't fix it

Environment structure

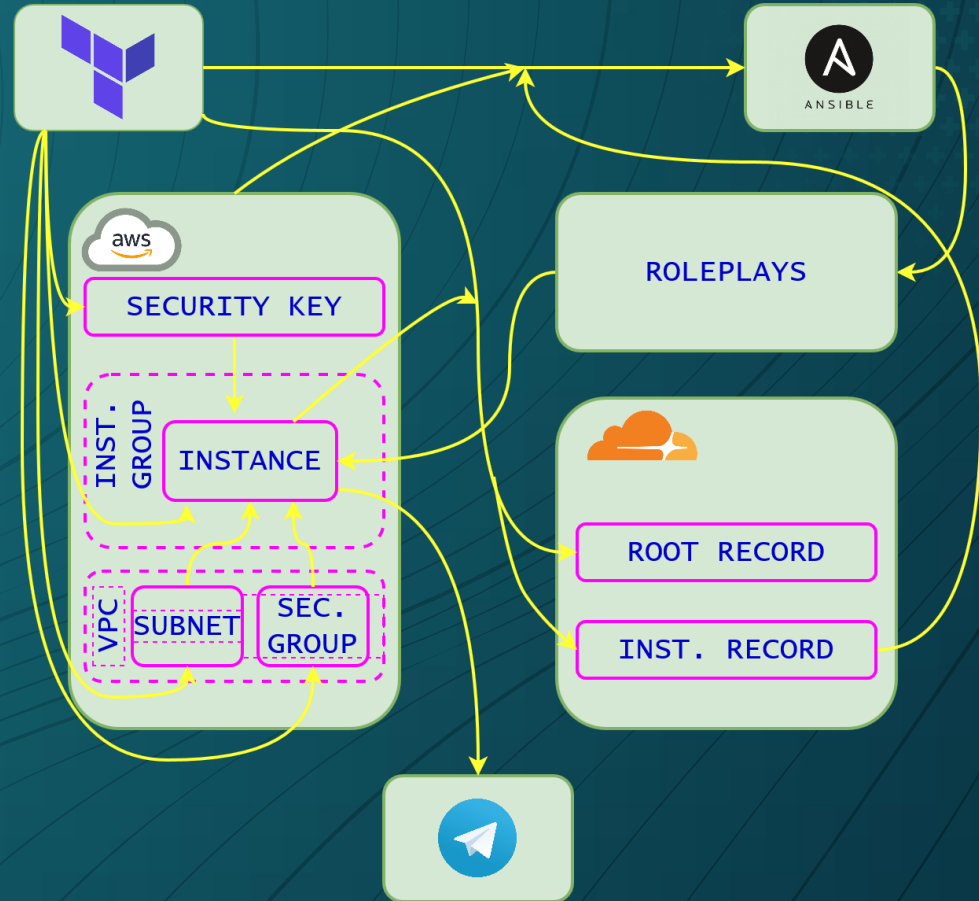
- Ubuntu server 18.04
- NGINX
- Tomcat app server
- MariaDB
- Letsencrypt client
- Telegraf



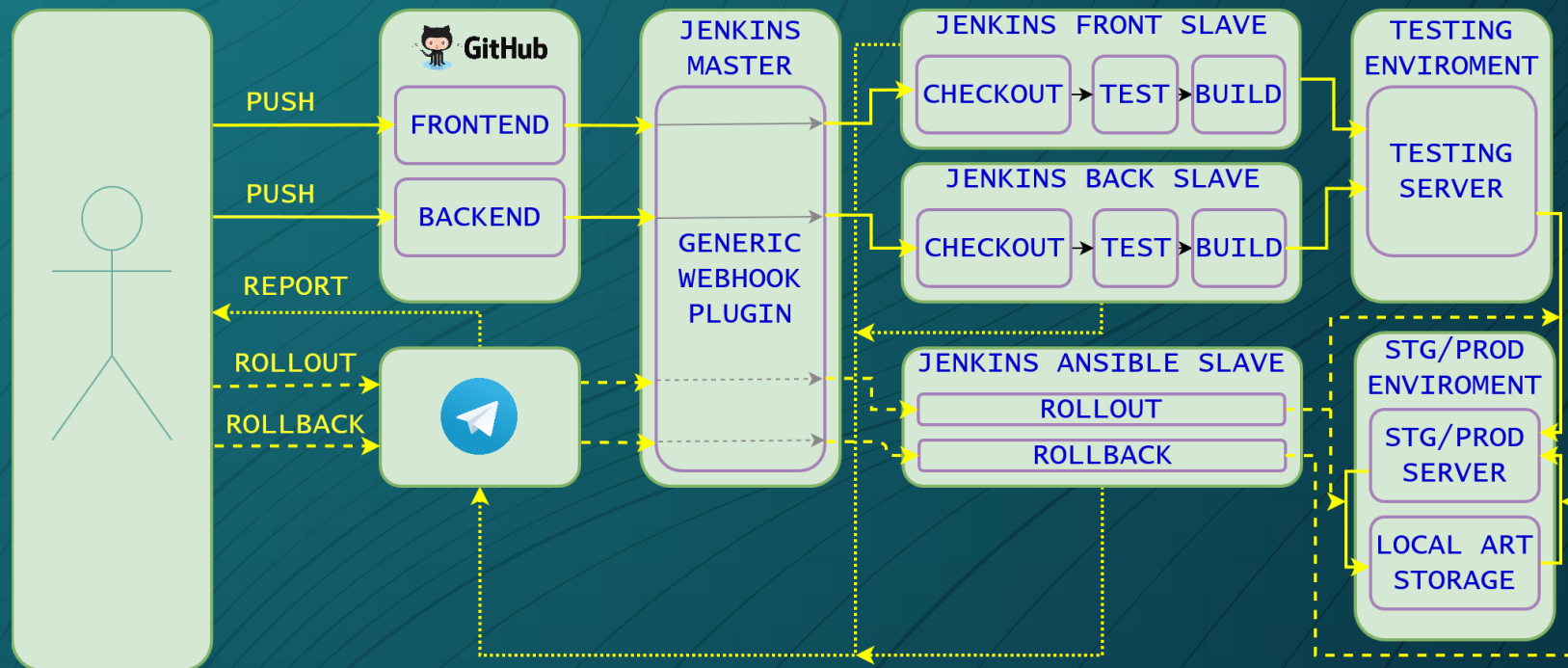
High availability design



Platform architecture



CI/CD structure



◆ Pros & Cons

- + Acceptable flexibility
- + Minimum application reworking
- + Optimization for full-ops maintenance
- A lot of work in CI/CD and platform development
- Low failover protection
- Using large amount of computing power

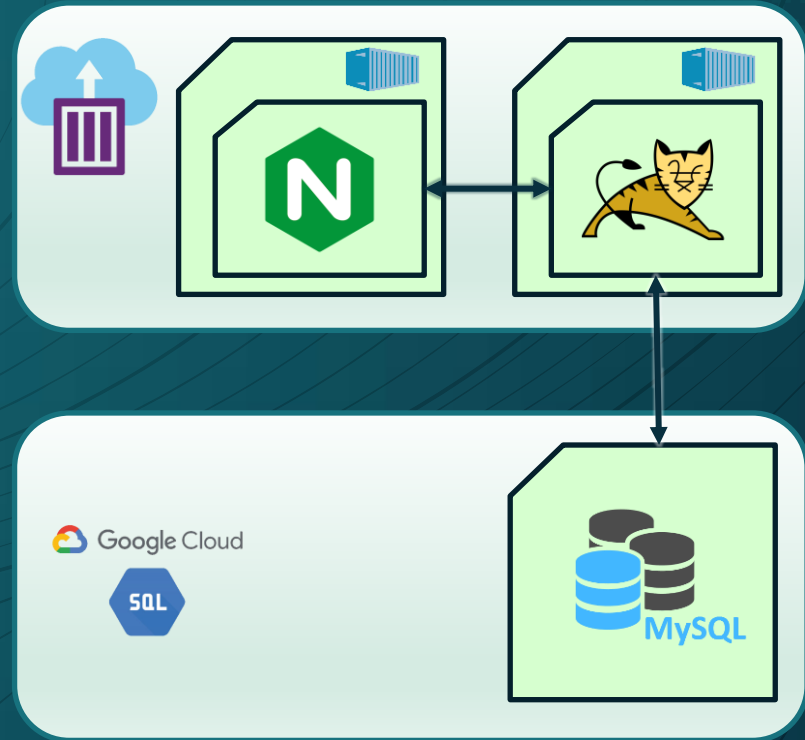


Quick-deployment solution

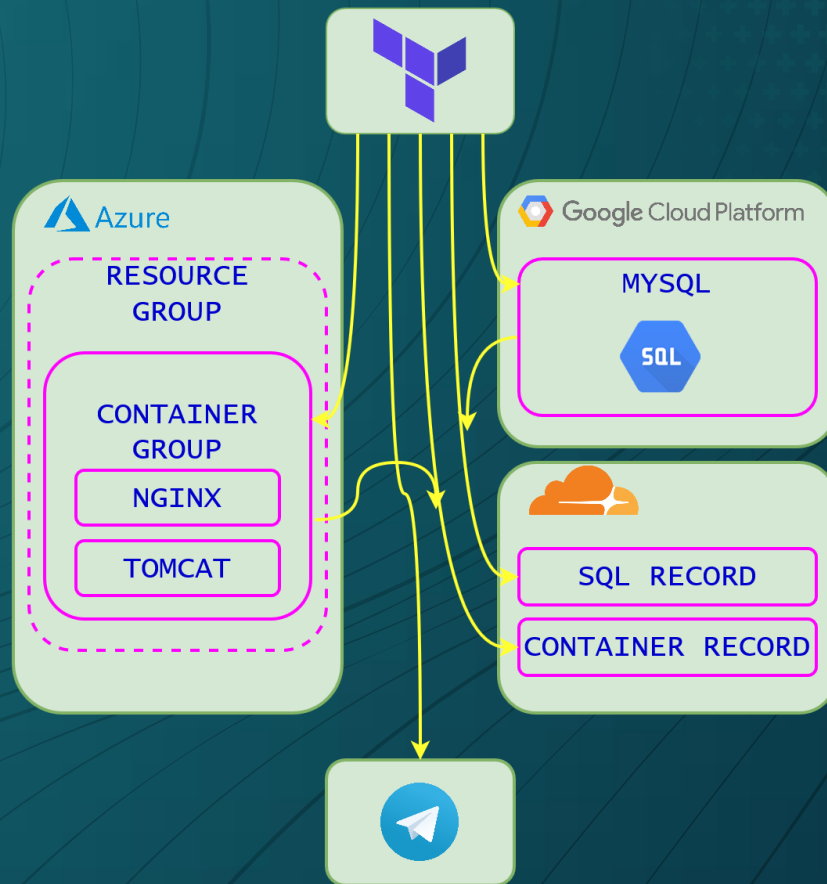
There is no time like the present

Environment structure

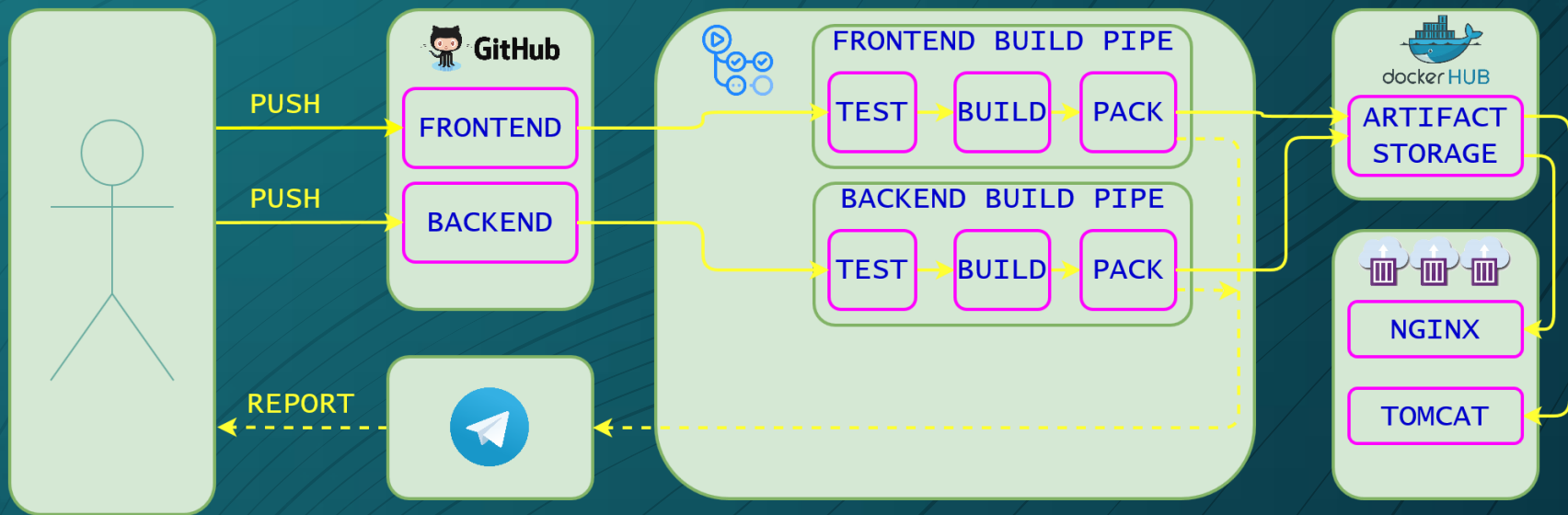
- Nginx container
- Tomcat container
- Azure container instances
- Google Cloud SQL



Platform architecture



CI/CD structure



◆ Pros & Cons

- + Fast pipeline development and deployment
- + Acceptable application build and deployment
- Bad security
- Bad scalability
- Bad availability optimization
- Application reworking needed
- Using third-party services

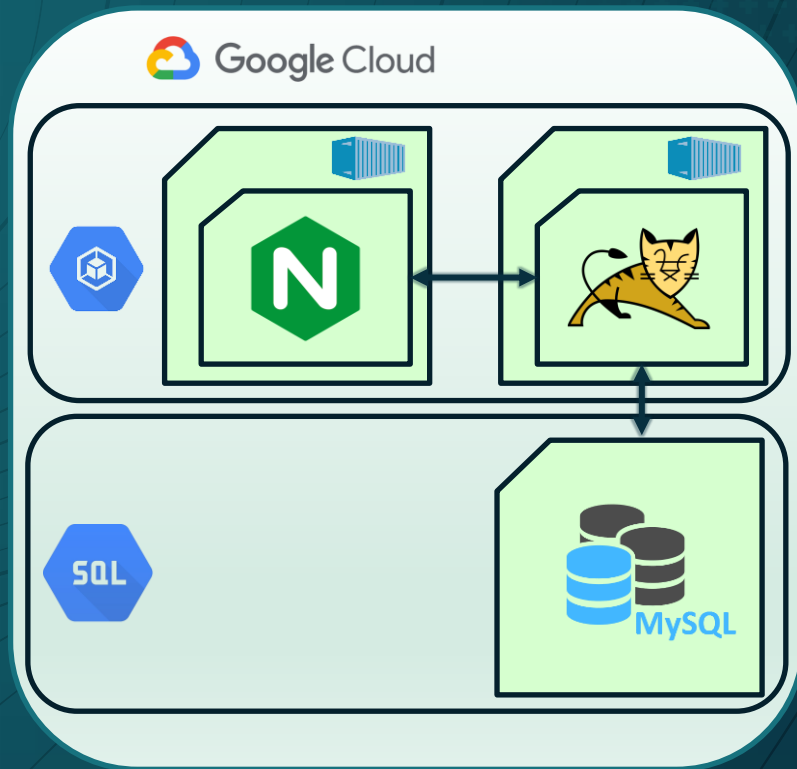


Enterprise solution

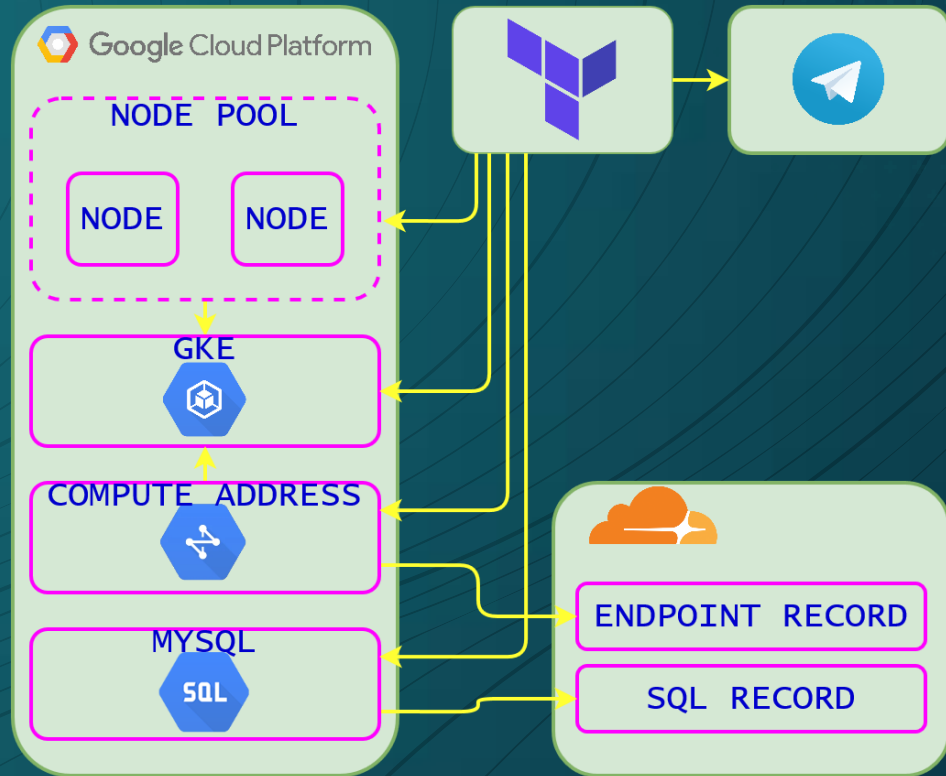
If a job's worth doing, it's worth doing well

Environment structure

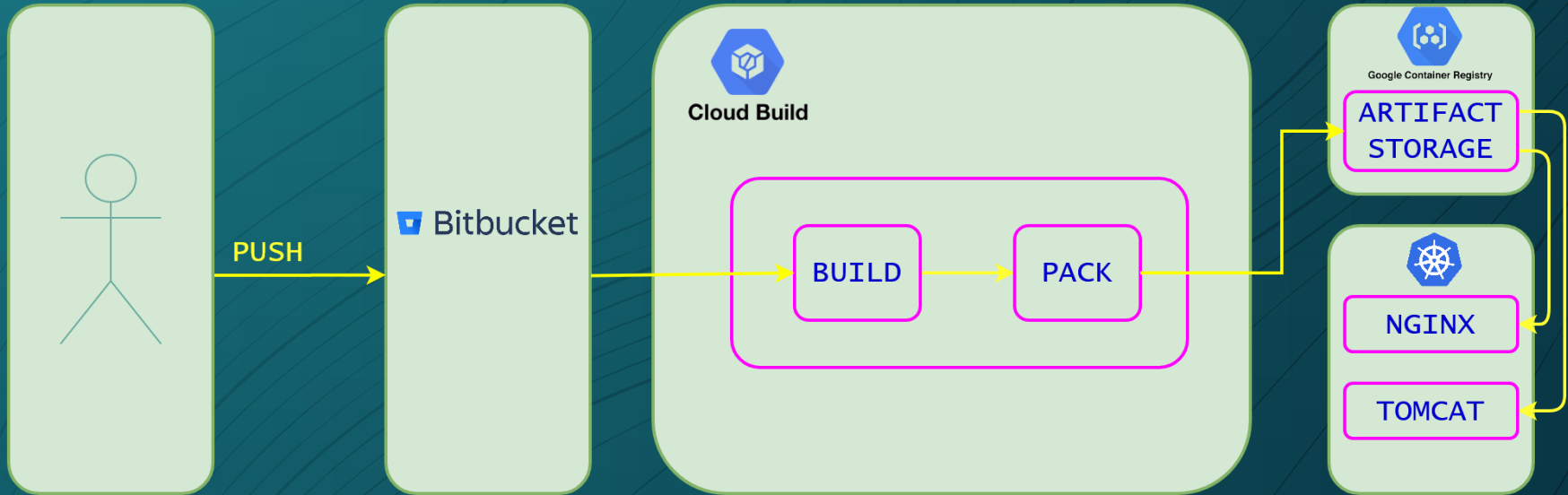
- Nginx container
- Tomcat container
- Kubernetes
- Google kubernetes engine
- Google Cloud SQL



Platform architecture



CI/CD structure



→ Pros & Cons

- + Hight availability
- + Hight scalability
- + Hight flexibility
- High complexity
- Hard to migrate
- Application reworking needed

◆ Conclusions

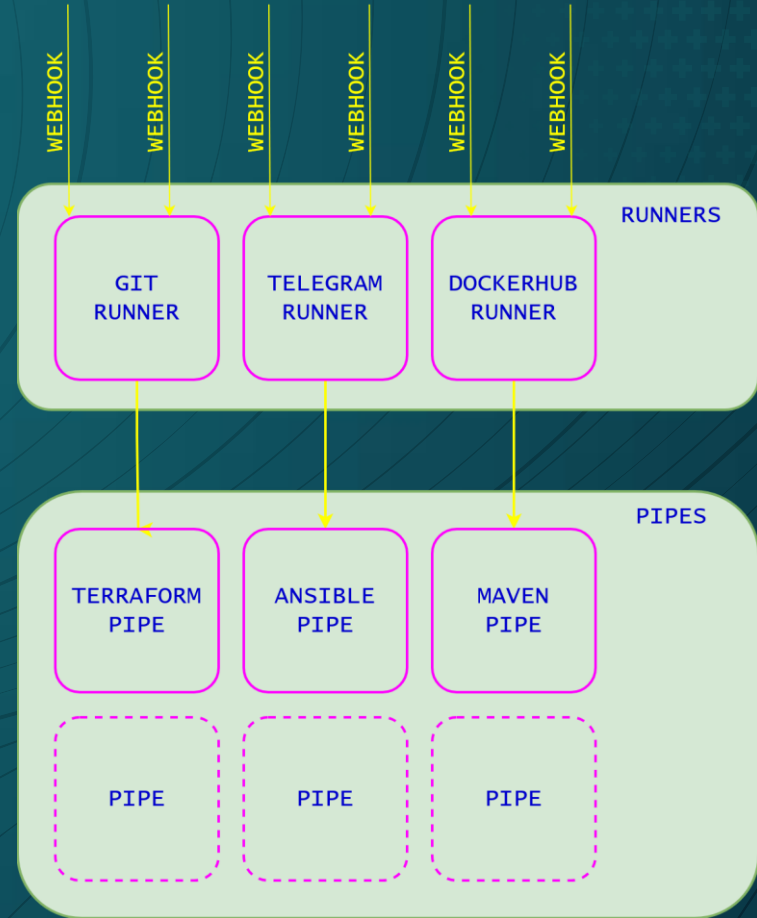
- CI/CD is integral in software building and deployment: smaller teams, constant changes, fast and real-time feedback, and app deployment. It provides benefits to product owners, development teams, and end-users
- Building CI/CD pipelines is complex task with many parameters. Using right CI/CD tools helps track and support the development process
- New inventions and implementations of CI/CD or platform design should be examined for pros and cons



Additional tips

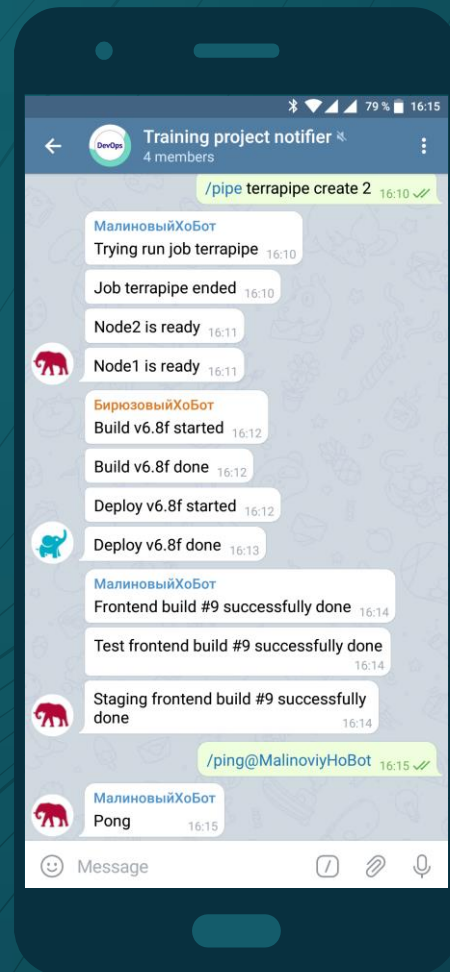
Nothing succeeds like success

◆ Pipeline bridges



Telegram chat

Operations and notifications!





Thanks!

You can mail me at ihor.porokhnia@gmail.com

Q & A session

??
??
??

