# The Journey to Implement CI/CD in Higher Education Institutions

The Story Behind Implement CI/CD in Higher Education Institutions

Wahyuni Puji - DevOps
Wahyuni.puji@uii.ac.id

## What is CI/CD

“Continuous Integration is a software development practice where members of a team integrate their work frequently; usually each person integrates at least daily – leading to multiple integrations per day.” --Martin Fowler

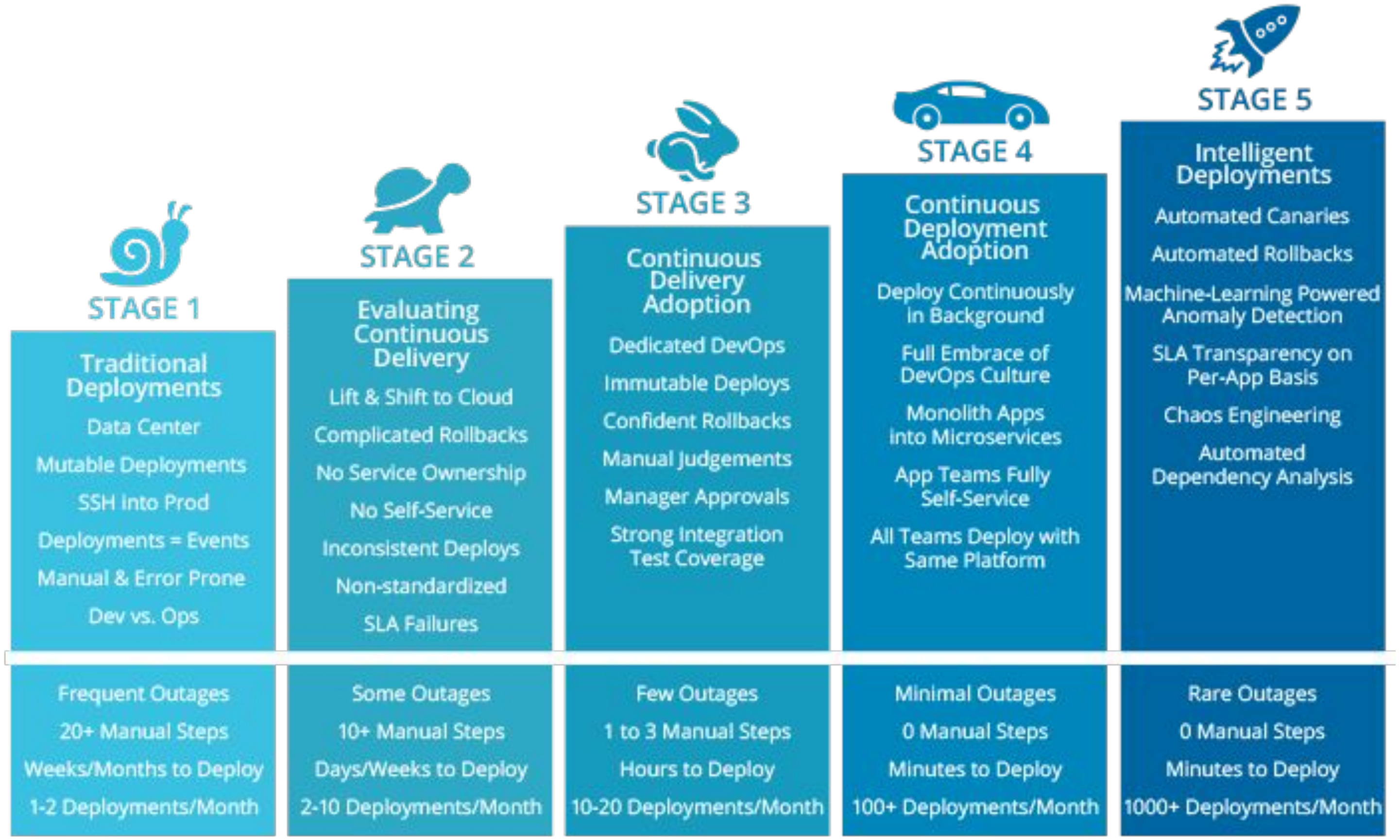
“Continuous Delivery is a software development discipline where you build software in such a way that the software can be released to production at any time” --Martin Fowler

Continuous Deployment is a third term that’s sometimes confused with Continuous Delivery. Where Continuous Delivery provides a process to create frequent releases but not necessarily deploy them, Continuous Deployment means that every change you make automatically gets deployed through the deployment pipeline.

armory

## STAGES OF SOFTWARE DELIVERY EVOLUTION

INDONESIA
OpenInfra Days



| STAGE 1 | STAGE 2 | STAGE 3 | STAGE 4 | STAGE 5 |
|---|---|---|---|---|
| Traditional Deployments | Evaluating Continuous Delivery | Continuous Delivery Adoption | Continuous Deployment Adoption | Intelligent Deployments |
| Data Center | Lift & Shift to Cloud | Dedicated DevOps | Deploy Continuously in Background | Automated Canaries |
| Mutable Deployments | Complicated Rollbacks | Immutable Deploys | Full Embrace of DevOps Culture | Automated Rollbacks |
| SSH into Prod | No Service Ownership | Confident Rollbacks | Monolith Apps into Microservices | Machine-Learning Powered Anomaly Detection |
| Deployments = Events | No Self-Service | Manual Judgements | App Teams Fully Self-Service | SLA Transparency on Per-App Basis |
| Manual & Error Prone | Inconsistent Deploys | Manager Approvals | All Teams Deploy with Same Platform | Chaos Engineering |
| Dev vs. Ops | Non-standardized | Strong Integration Test Coverage | | Automated Dependency Analysis |
| | SLA Failures | | | |
| Frequent Outages | Some Outages | Few Outages | Minimal Outages | Rare Outages |
| 20+ Manual Steps | 10+ Manual Steps | 1 to 3 Manual Steps | 0 Manual Steps | 0 Manual Steps |
| Weeks/Months to Deploy | Days/Weeks to Deploy | Hours to Deploy | Minutes to Deploy | Minutes to Deploy |
| 1-2 Deployments/Month | 2-10 Deployments/Month | 10-20 Deployments/Month | 100+ Deployments/Month | 1000+ Deployments/Month |

Biznet GioCloud Biznet Mellanox TECHNOLOGIES BANK BRI OSF OpenStack Foundation

## Why CI/CD

- Ensures changes to code base are properly tracked, tested, and built
- Automation! Lessens chance of human error
- Easily track source of bugs and ability to rollback
- Faster time to market
- Avoid outages from deployments
- Happier development & operations teams
- More metrics to review and act on

INDONESIA
OpenInfra Days

## Why we Implement CI/CD

Biznet GioCloud Biznet Mellanox TECHNOLOGIES BANK BRI OSF OpenStack Foundation

INDONESIA
OpenInfra Days

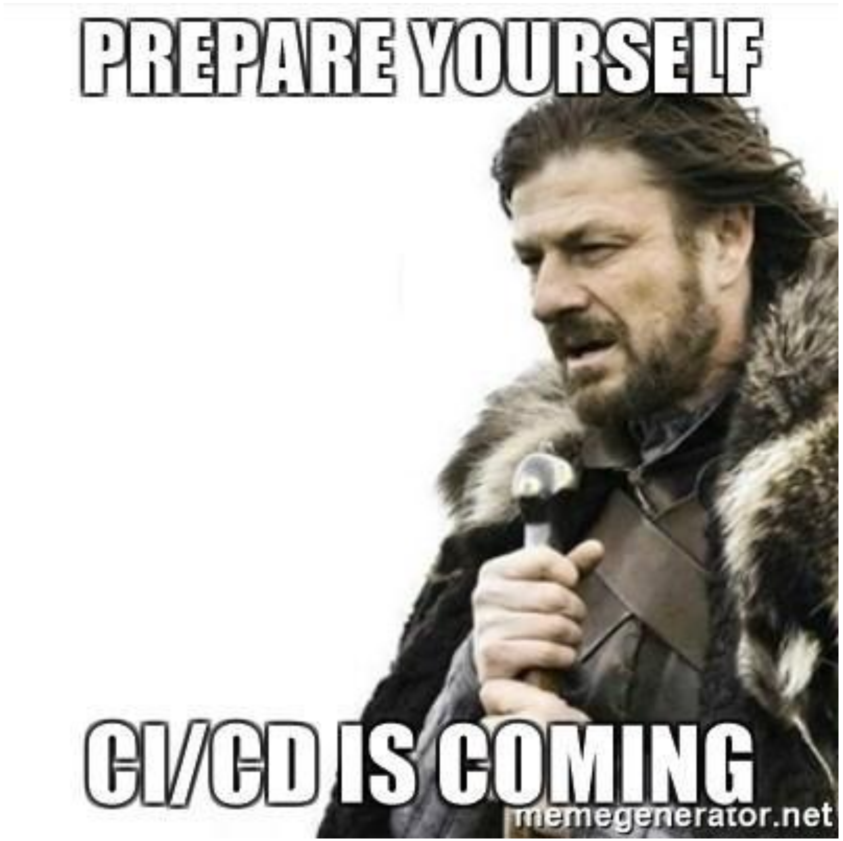
I NEED TO KNOW WHY MOVING OUR APP TO THE CLOUD DIDN'T AUTOMATICALLY SOLVE ALL OUR PROBLEMS.
YOU WOULDN'T LET ME RE-ARCHITECT THE APP TO BE CLOUD-NATIVE.
JUST PUT IT IN CONTAINERS.
YOU CAN'T SOLVE A PROBLEM JUST BY SAYING TECHY THINGS.
KUBERNETES.
Dilbert.com @ScottAdamsSays
11-08-17 © 2017 Scott Adams, Inc./Dist. by Andrews McMeel

Distributed system.

INDONESIA
OpenInfra Days

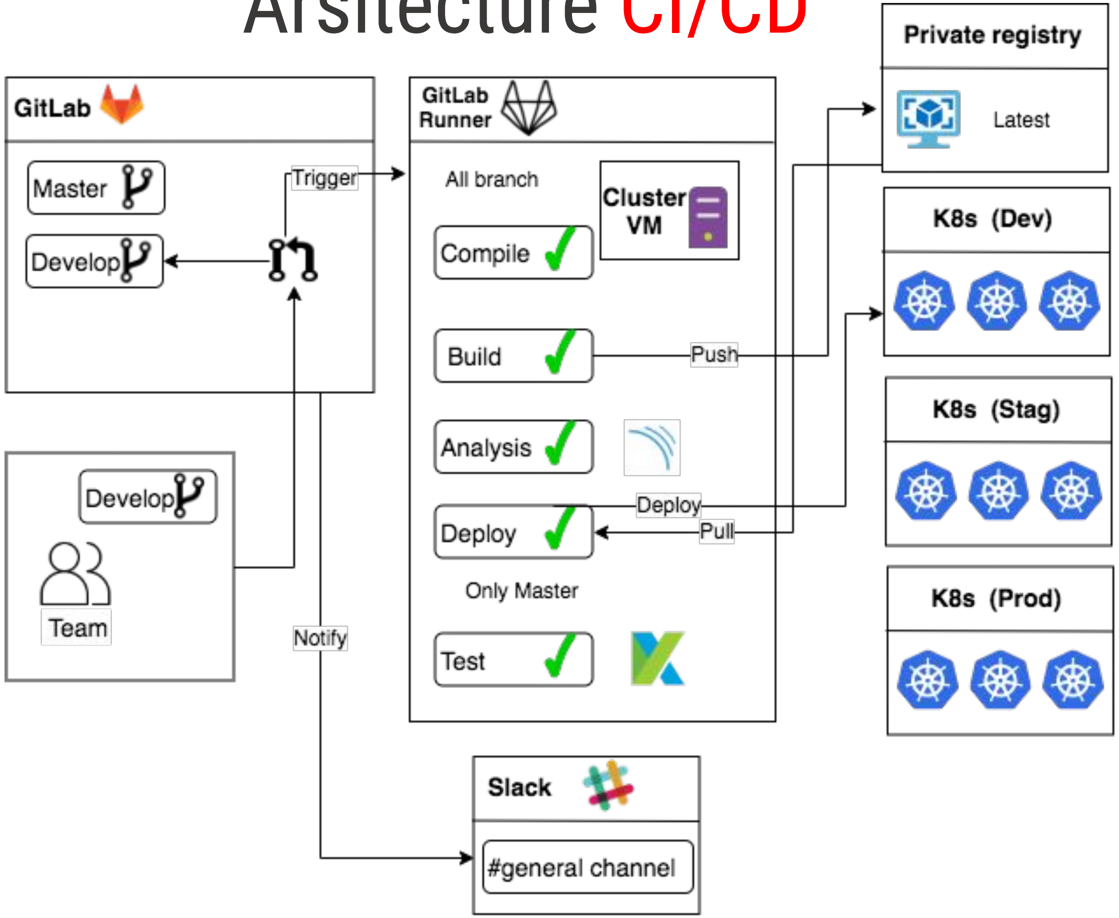
Biznet GioCloud Biznet Mellanox TECHNOLOGIES BANK BRI OSF OpenStack Foundation

## How we Adopted ?



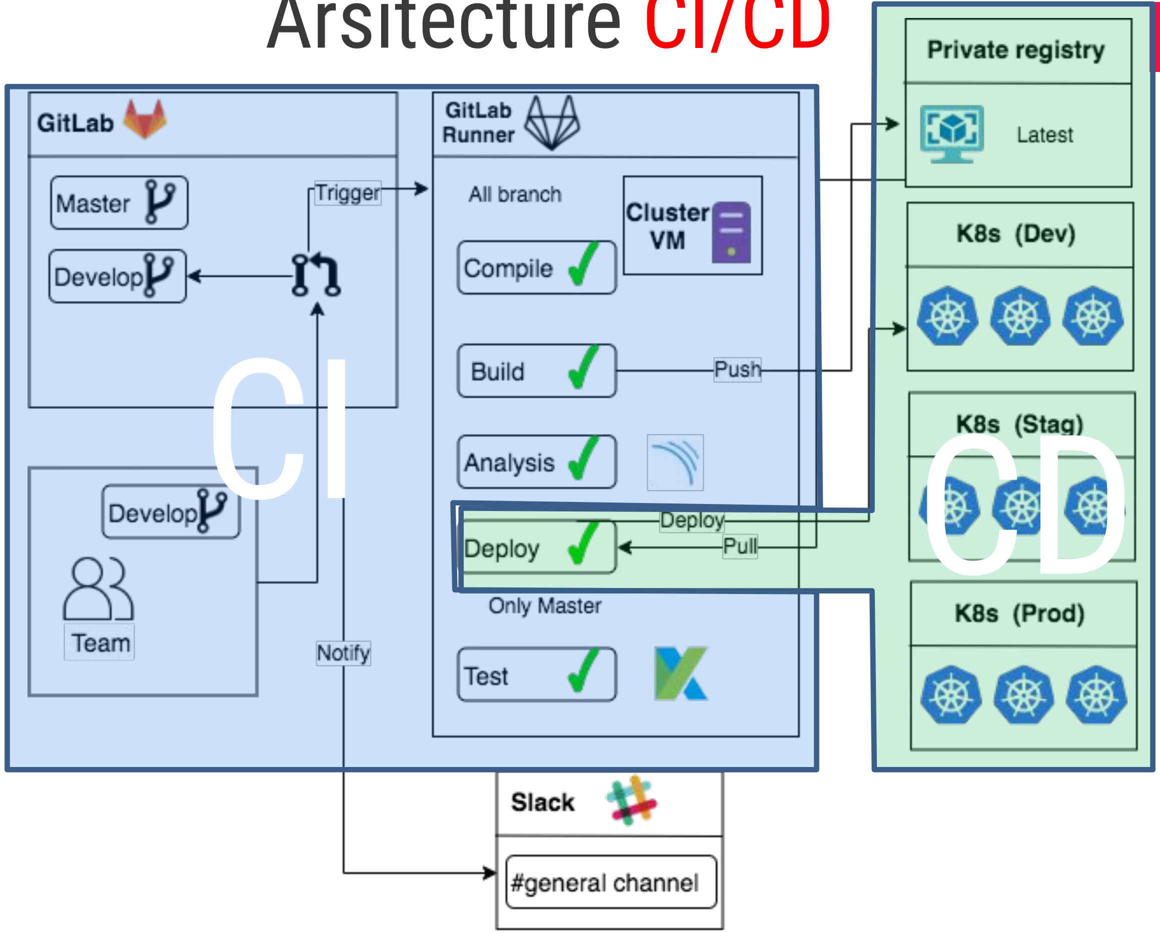
# Arsitecture CI/CD

- Tools that we use
- Infrastructure

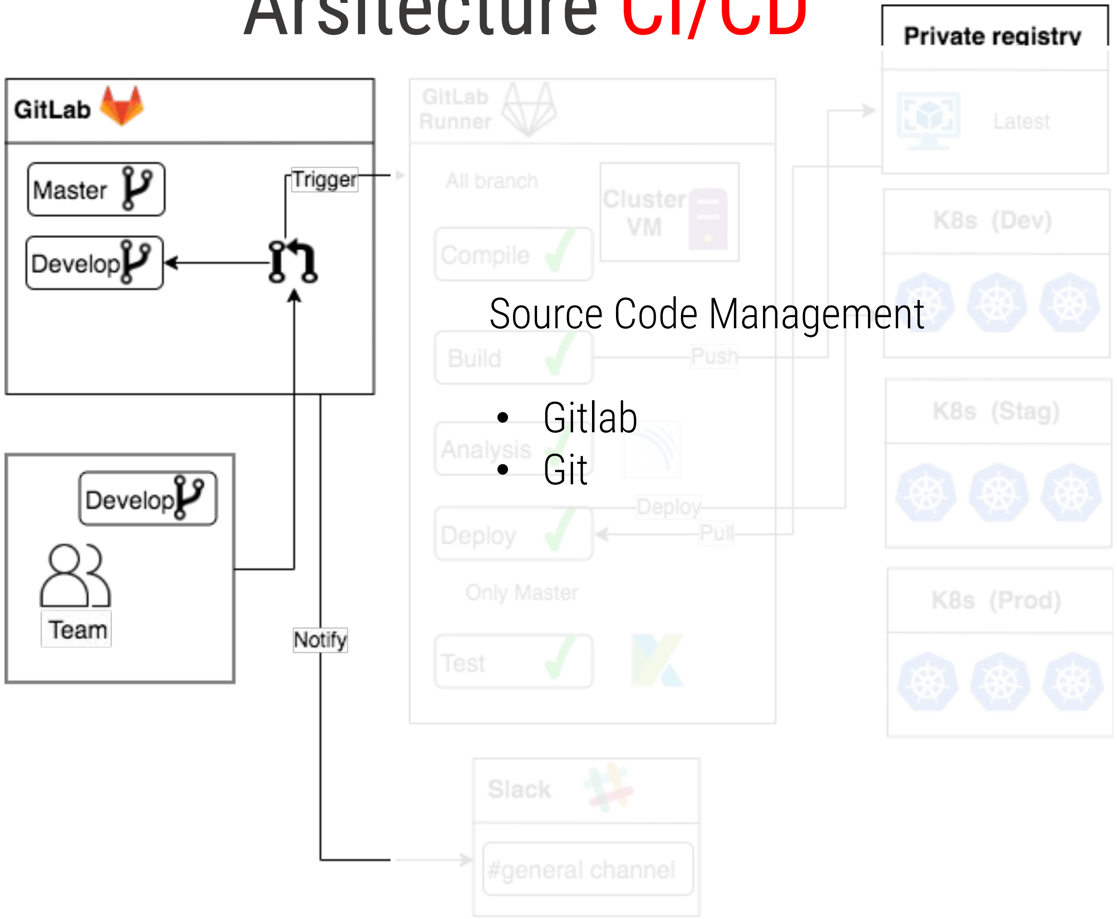


# Arsitecture CI/CD

- Tools that we use
- Infrastructure



# Arsitecture CI/CD



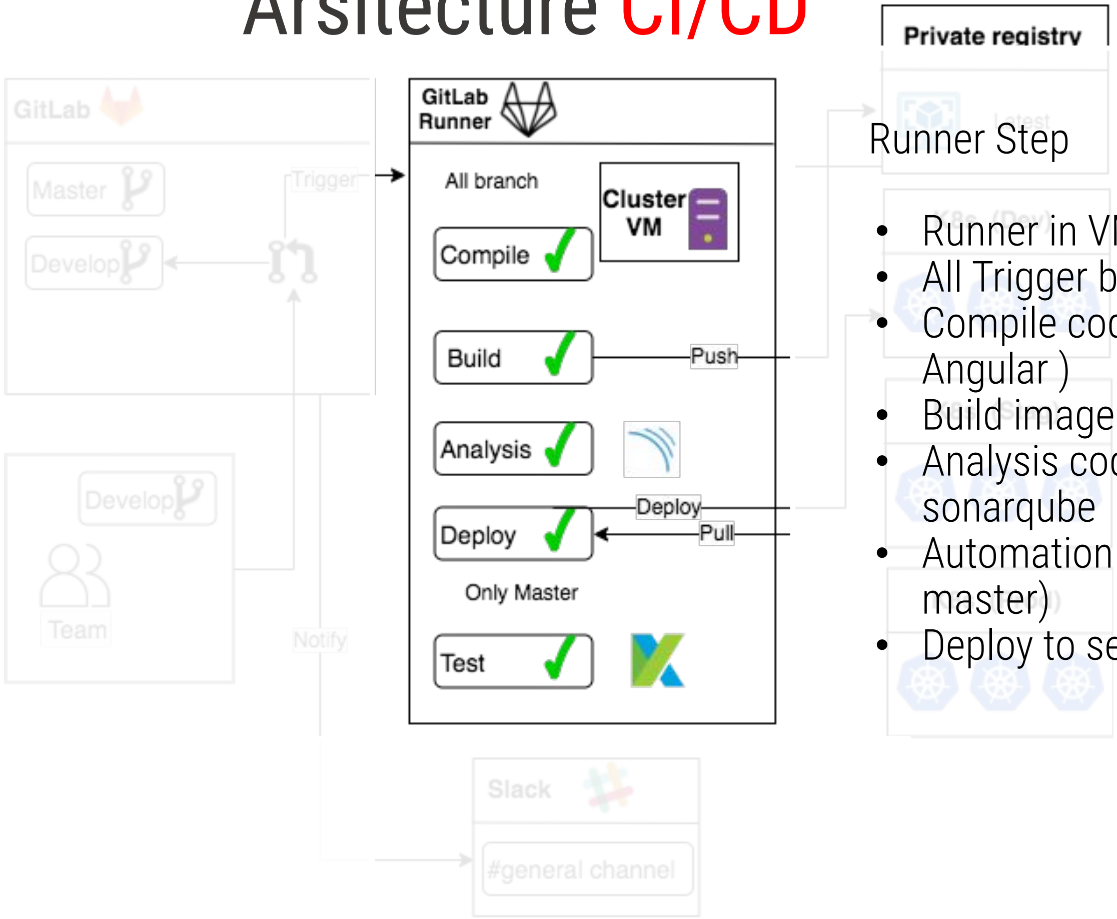
- Tools that we use
- Infrastructure

Source Code Management

- Gitlab
- Git

# Arsitecture CI/CD

- Tools that we use
- Infrastructure



Runner Step

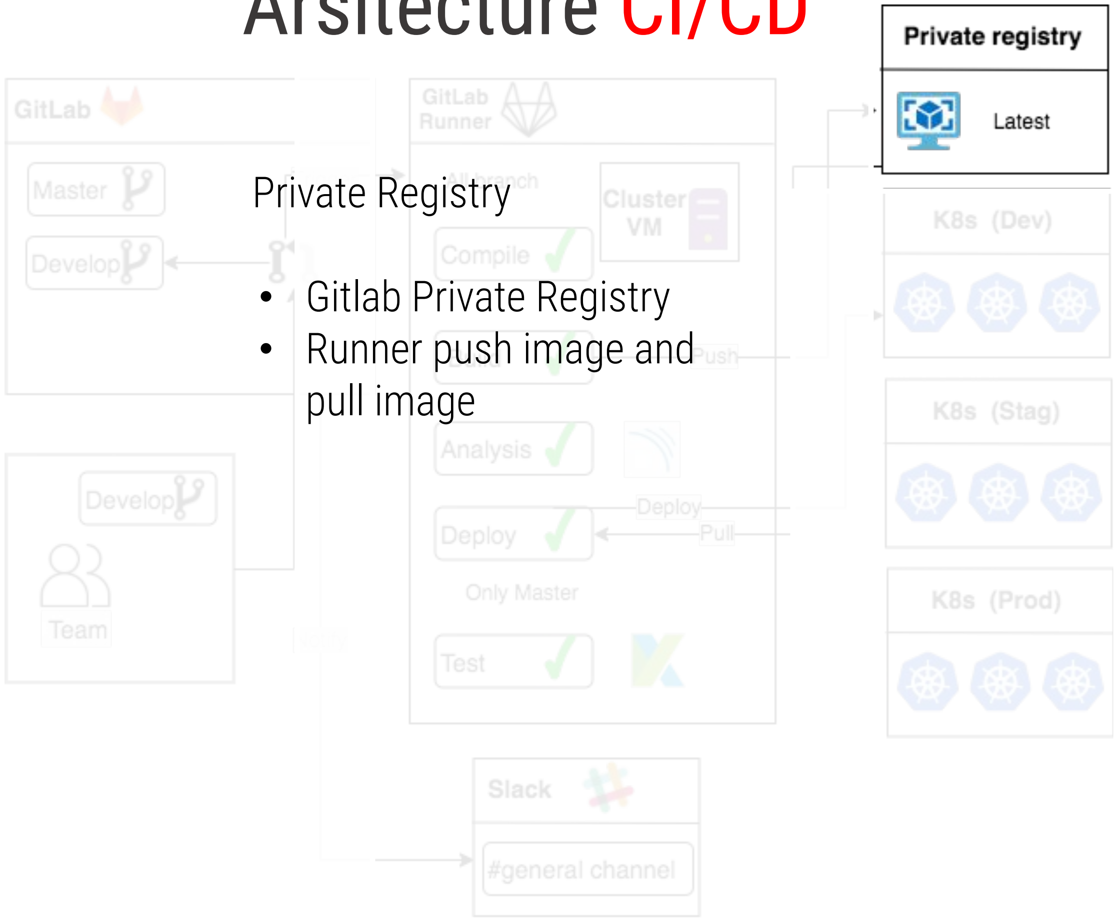
- Runner in VM
- All Trigger by tag
- Compile code (Go, Java, Angular )
- Build image
- Analysis code with sonarqube
- Automation testing (only master)
- Deploy to server

# Arsitecture CI/CD

- Tools that we use
- Infrastructure

Private Registry

- Gitlab Private Registry
- Runner push image and pull image

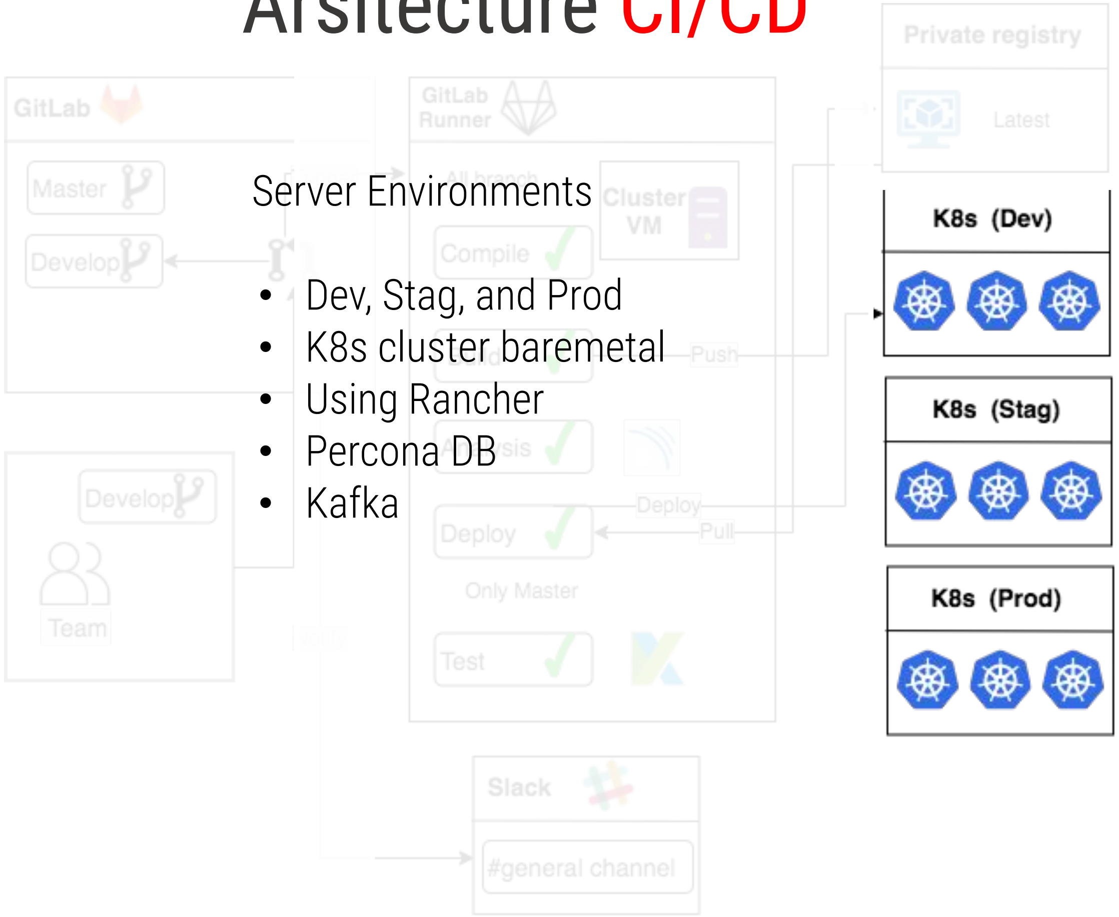


# Arsitecture CI/CD

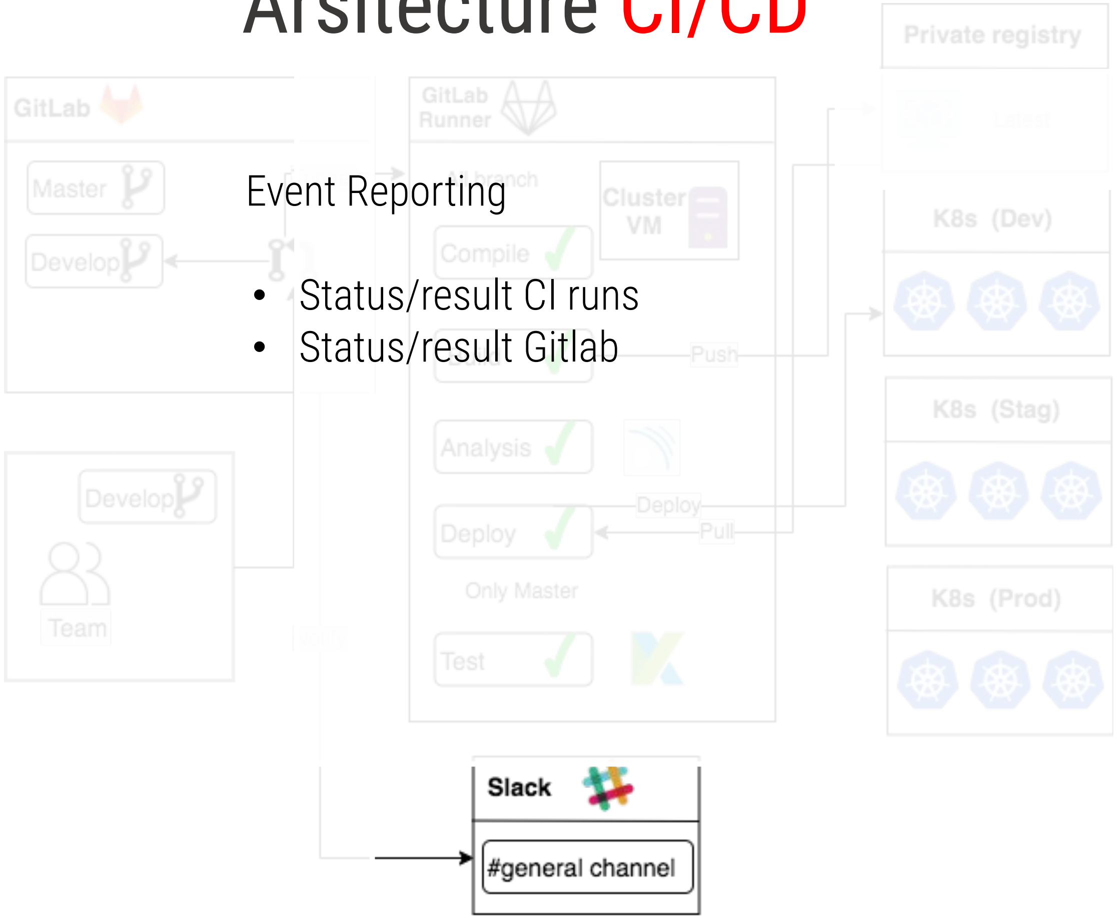
- Tools that we use
- Infrastructure

Server Environments

- Dev, Stag, and Prod
- K8s cluster baremetal
- Using Rancher
- Percona DB
- Kafka



# Arsitecture CI/CD



Event Reporting

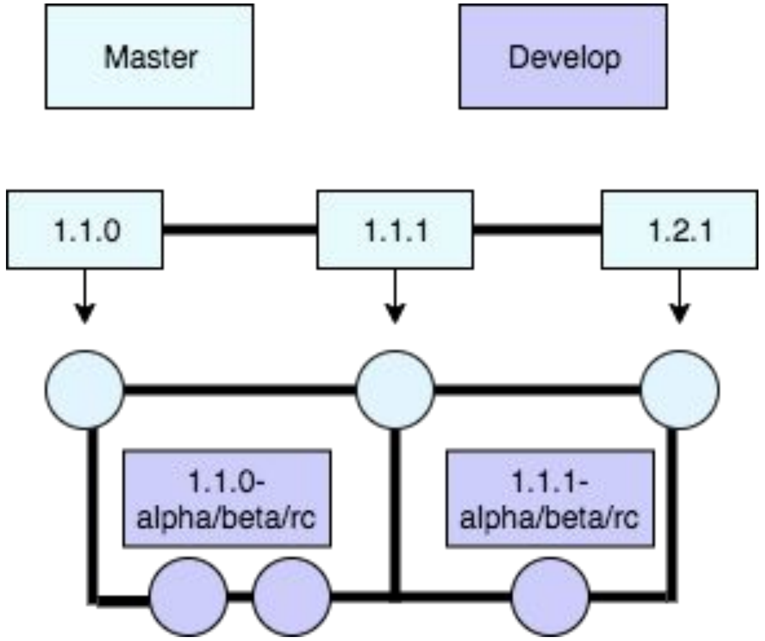
- Status/result CI runs
- Status/result Gitlab

- Tools that we use
- Infrastructure

# Git Flow

## Git Flow

- Versioning
- Branching
- Trigger



## Is that Enough?

## New Problem

01 Runner gitlab are used interchangeably (need more time)

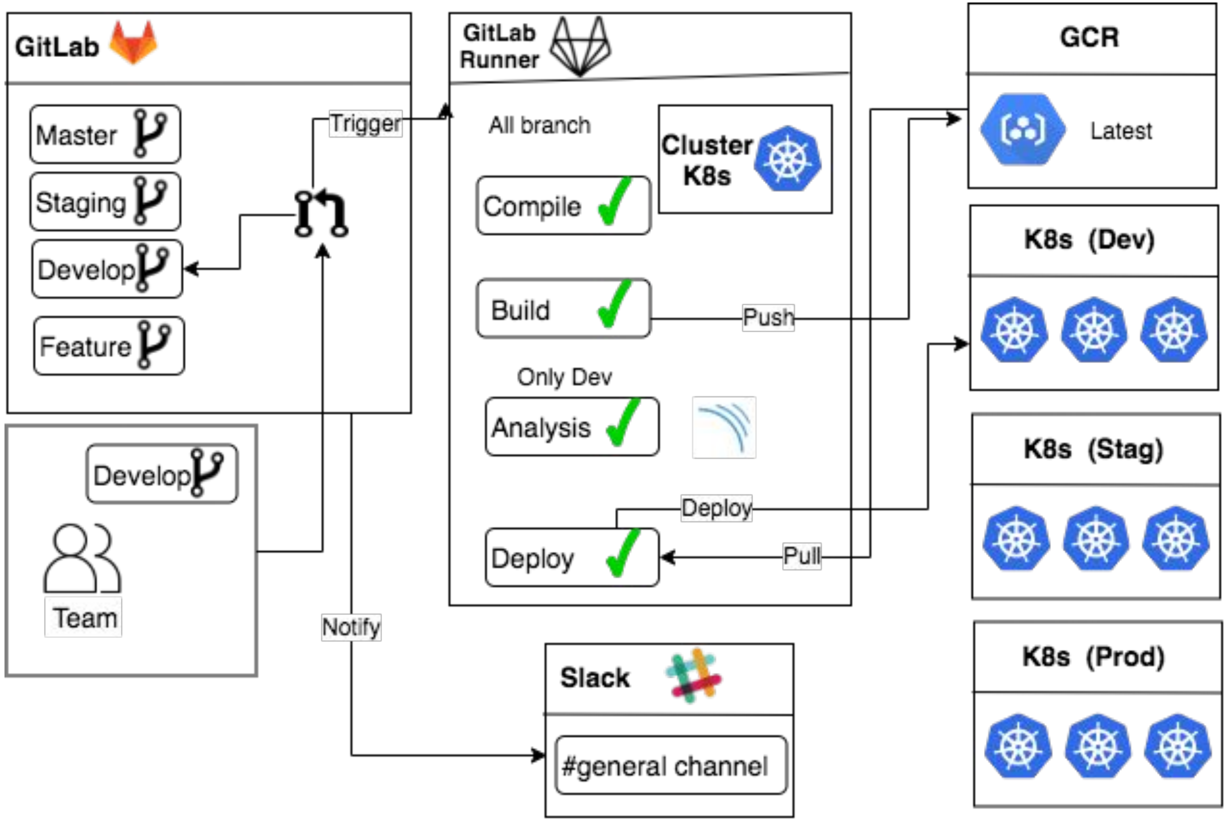
02 More tools, make developer confused

03 Not every automated test can meet the needs

# UPGRADE

- Fix Git flow
- Upgrade the infrastructure runner to auto scaling with k8s
- Minimalistic tools integrated into CI / CD

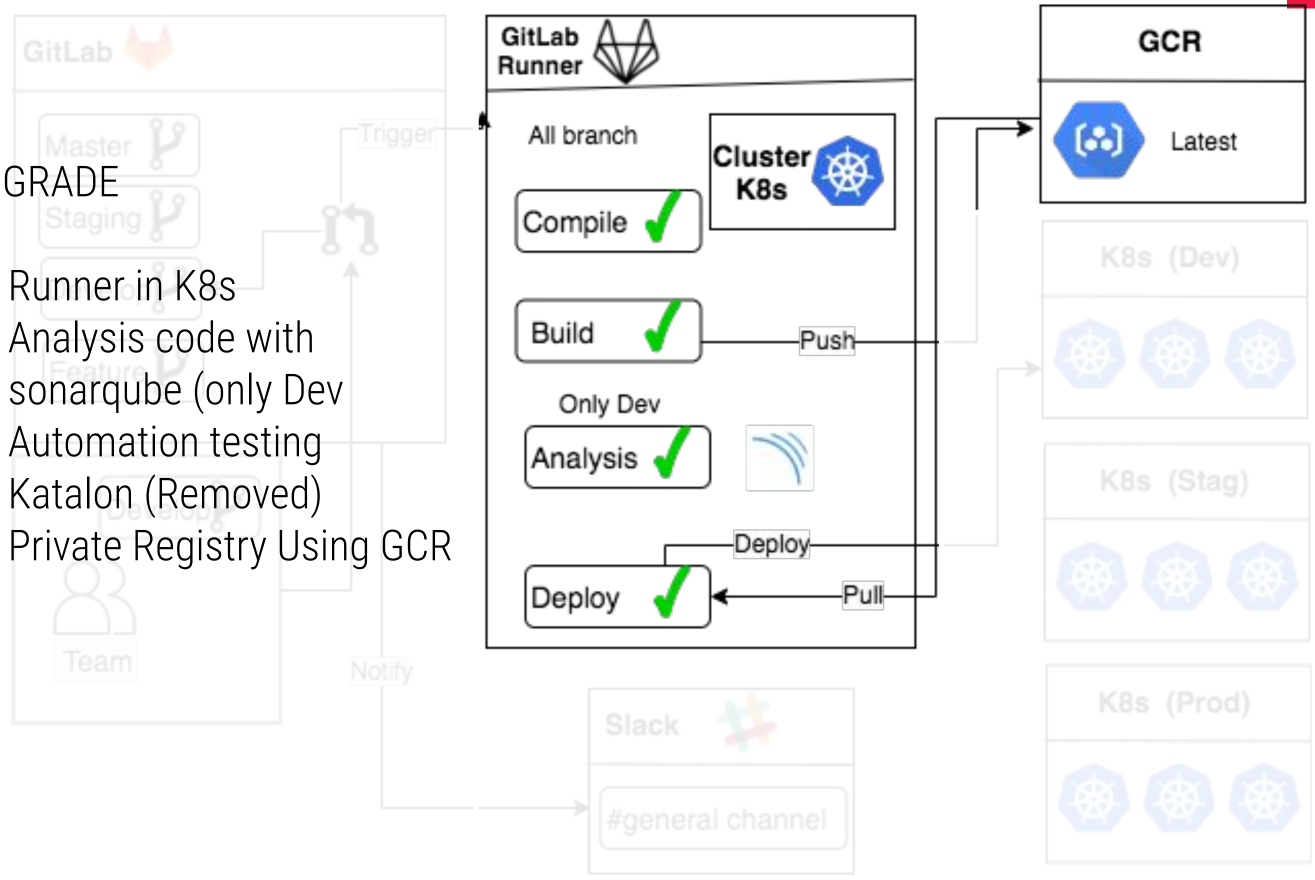
# New Arsitecture CI/CD



# New Arsitecture CI/CD

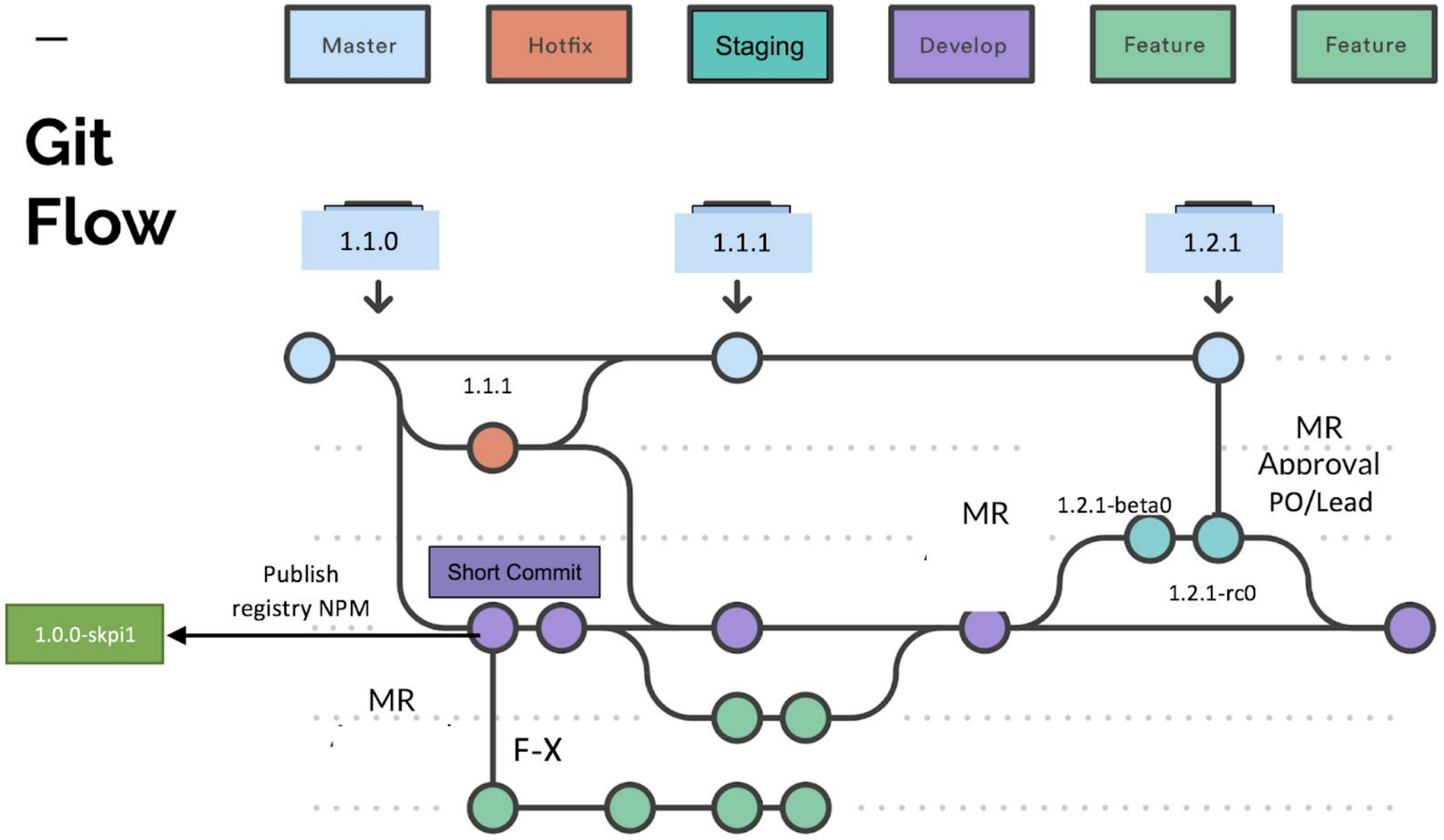
UPGRADE

- Runner in K8s
- Analysis code with sonarqube (only Dev
- Automation testing Katalon (Removed)
- Private Registry Using GCR



—

## Git Flow


Master
Hotfix
Staging
Develop
Feature
Feature
1.1.0
1.1.1
1.2.1
1.1.1
MR
Approval
PO/Lead
MR
1.2.1-beta0
1.2.1-rc0
Short Commit
Publish
registry NPM
1.0.0-skpi1
MR
F-X

INDONESIA
OpenInfra Days

# The Problem is Resolved

But.....

Biznet GioCloud Biznet Mellanox TECHNOLOGIES BANK BRI OSF OpenStack Foundation

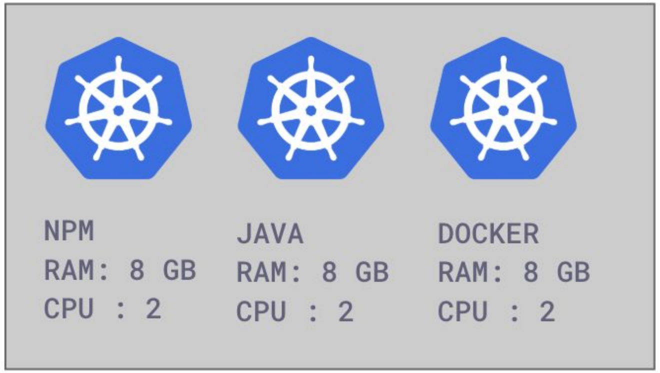
Angular

Java

PHP

Golang

Need more resources for runner



NPM
RAM: 8 GB
CPU : 2

JAVA
RAM: 8 GB
CPU : 2

DOCKER
RAM: 8 GB
CPU : 2

Cluster k8s for runner Gitlab

RAM: 70 GB, CPU : 25

IF we have 10 runner active in same time, the resources cluster not enough. (kubectl error) → need to restart

INDONESIA
*OpenInfra Days*

## Next Planning

**01** Make new infrastructure runner with various categories, not only global

**02** Implement bazel to reduce pipeline compile in runner

**03** Try to implement continuous testing

INDONESIA
OpenInfra Days

## Conclusion

- Chosen right tools
- more patient
- Give learning not only for developer, and superior
- If the process is painful, you're doing it wrong
- Make your whole team come on board before starting to adopt continuous integration.
- Integrating CI Into Your Existing Development Flow
- Creating Fear-Driven Development
- Developers Ignoring Error Messages
- Keep Learning, Keep Upgrading

Biznet GioCloud Biznet Mellanox TECHNOLOGIES BANK BRI OSF OpenStack Foundation

GIT PUSH
AND RUN
memegenerator.net

Thank you!