Deploy Multinode GitLab Runner in openSUSE® 15.1 Instances with Ansible Automation

DevOps Practice in openSUSE

Samsul Ma'arif – DevOps Engineer
samsul@nacita.id

Hi, it is me……

Samsul Ma’arif

DevOps Engineer @ DOT Indonesia

samsul@nacita.id

085747526846

https://www.linkedin.com/in/samsulmaarif

Community & Affiliation

KOMUNITAS GNU/LINUX MALANG

Kabar Linux
kabar seputar linux dan open source

BlankOn

Na Cita

NOLSATU

OPENSTACK USER GROUP INDONESIA

GNOME INDONESIA

openSUSE.ID

Keyword

CI/CD, Docker, Ansible, GitLab CI, openSUSE, Leap, Deployment

Outline

- Short Introduction
  - GitLab & GitLab runner
  - CI/CD overview
  - Ansible
- Schenario
- Demo

About GitLab

- A single application for the entire DevOps lifecycle
- Open source project
- Written in Rails
- Continous Integration (CI/CD)
- Source Code Management
- Auto DevOps
- Agile Development

https://about.gitlab.com

GitLab Features

Manage	Plan	Create	Verify	Package	Secure	Release	Configure	Monitor	Defend
Since 2016 GitLab added:	**Since 2011 GitLab added:**	**Since 2011 GitLab added:**	**Since 2012 GitLab added:**	**Since 2016 GitLab added:**	**Since 2017 GitLab added:**	**Since 2016 GitLab added:**	**Since 2018 GitLab added:**	**Since 2017 GitLab added:**	**Since 2019 GitLab added:**
Audit Management	Issue Tracking	Source Code Management	Continuous Integration (CI)	Package Registry	SAST	Continuous Delivery	Auto DevOps	Metrics	WAF
Authentication and Authorization	Kanban Boards	Code Review	Code Quality	Container Registry	DAST	Release Orchestration	Kubernetes Configuration	Logging	**On our roadmap:**
DevOps Score	Time Tracking	Design Management	Web Performance	Dependency Proxy	Secret Detection	Pages	ChatOps	Tracing	RASP
Value Stream Management	Agile Portfolio Management	Wiki	Usability Testing		Dependency Scanning	Review apps	Runbooks	Cluster Monitoring	Threat Detection
	Service Desk	Web IDE			Container Scanning	Incremental Rollout	Serverless	Error Tracking	UEBA
		Snippets			License Compliance	Feature Flags	Infrastructure as Code	Incident Management	Vulnerability Management
On our roadmap:	**On our roadmap:**		**On our roadmap:**	**On our roadmap:**	**On our roadmap:**	**On our roadmap:**	**On our roadmap:**	**On our roadmap:**	DLP
Code Analytics	Requirements Management	**On our roadmap:**	Load Testing	Helm Chart Registry	IAST	Release Governance	Chaos Engineering	Synthetic Monitoring	Storage Security
Workflow Policies	Quality Management	Live Coding	System Testing	Dependency Firewall	Fuzzing	Secrets Management	Cluster Cost Optimization	Status Page	Container Network Security

What is CI/CD?

Continuous Integration is the practice of integrating code into a shared repository and building/testing each change automatically, as early as possible - usually several times a day.

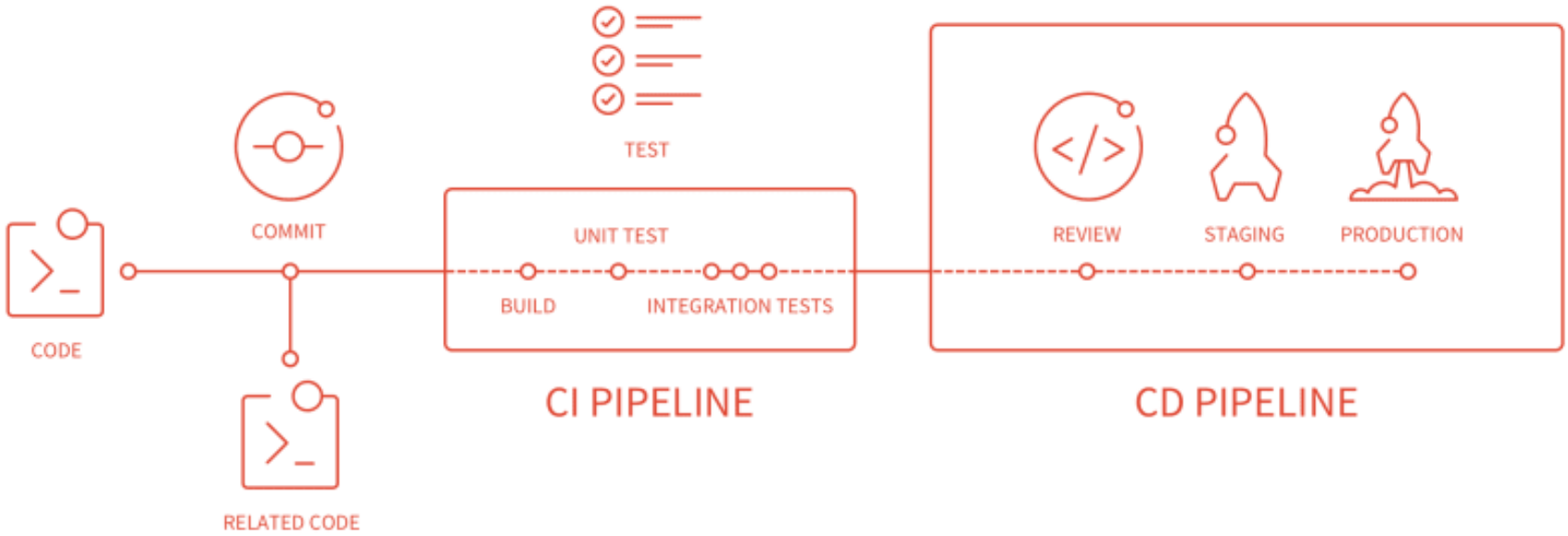
Continuous Delivery adds that the software can be released to production at any time, often by automatically pushing changes to a staging system.

Continuous Deployment goes further and pushes changes to production automatically.

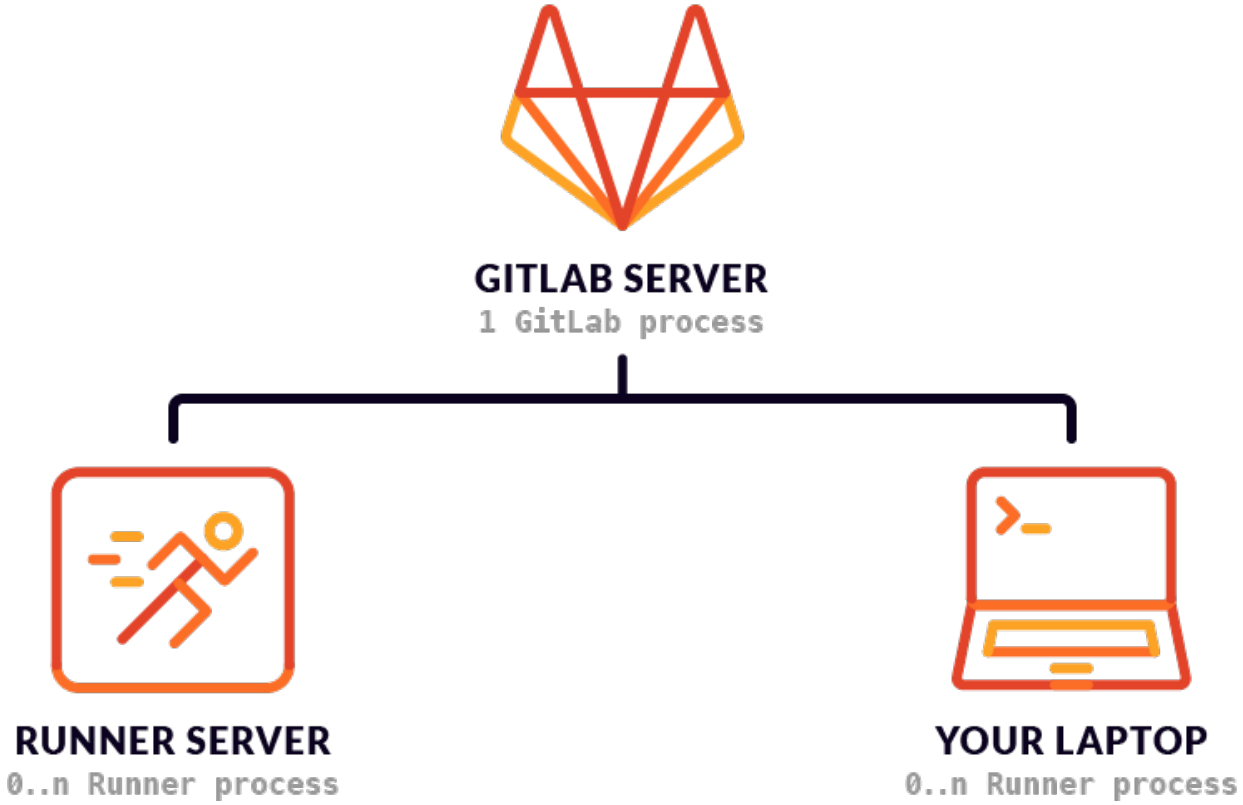
Why we need CI/CD?

- **Continuous Integration**
  - Detects errors as quickly as possible
  - Reduces integration problems
  - Avoid compounding problems
- **Continuous Delivery**
  - Ensures every change is releasable
  - Lowers risk of each release
  - Delivers value more frequently
  - Tight customer feedback loops

GitLab CI/CD Workflow



GitLab CI/CD Architecture



GitLab Runner

- Run pipeline jobs in GitLab
- run the code defined in .gitlab-ci.yml
- Written in Go
- no language specific requirements are needed
- designed to run on the GNU/Linux, macOS, and Windows operating systems

GitLab Runner Executor

- SSH
- Shell
- Parrallels
- VirtualBox
- Docker
- Docker Machine
- Kubernetes
- Custom

GitLab Runner Requirement

- It depends on:
  - The type of executor you configured on GitLab Runner.
  - Resources required to run build jobs.
  - Job concurrency settings.

GitLab Runner types

- Shared Runner
- Specific Runner
- Group Runner

GitLab Runner

- Run pipeline jobs in GitLab
- run the code defined in .gitlab-ci.yml
- Written in Go
- no language specific requirements are needed
- designed to run on the GNU/Linux, macOS, and Windows operating systems

Example .gitlab-ci.yml

.gitlab-ci.yml

```yaml
stages:
  - test
  - deploy

run_test:
  stage: test
  image: samsulmaarif/php-laravel:7.3
  allow_failure: false
  services:
    - mysql:5.7
  variables:
    MYSQL_DATABASE: project
    MYSQL_ROOT_PASSWORD: rahasia
    DB_HOST: mysql
    DB_USERNAME: root
  tags:
    - dot
  cache:
    paths:
    - vendor/
  script:
    - curl -sS https://getcomposer.org/installer | php
    - cp .env.testing .env
    - php composer.phar install
    - php artisan key:generate
    - php artisan migrate
```

Ansible Automation

- Open Source
- Configuration Management
- Deployment tools
- Written in Python
- Cross platform
- Orchestration

Ansible Automation

- Open Source
- Configuration Management
- Deployment tools
- Written in Python
- Cross platform
- Orchestration

Why Ansible?

- **Simple**, use syntax written in YAML called **playbooks**
- **Agentless**
- **Powerfull and Flexible**
- **Efficient**

Ansible Modules

- Module control system resource, packages, files, or nearly anything else
- Over 450 ship with Ansible
- Enable regular users to easily work with complex systems
- But more on this later

Install Ansible

- **OS Package**
  - Zypper
  - APT
  - EPEL
- **Also available from**
  - Pypi
  - Source (via GitHub)

Schenario

- A machine with > 16Gb of RAM or multiple machine with same or different specification
- We use that machine as a Runner
- Install openSUSE 15.1
- Install the runner on that machine
- Register the runner to GitLab instance
- Automate the installation and registration with Ansible

Demo time

12:00-12.30

openSUSE

Resource for demo

https://github.com/samsulmaarif/ansible-gitlab-runner

- Prepare the openSUSE 15.1 VM
- Clone the repo
- Follow the tutorial in README.md

References

- https://about.gitlab.com/product/continuous-integration/
- https://docs.ansible.com
- https://docs.gitlab.com
-

Join the conversation,
contribute & have a lot of fun!

www.opensuse.org

openSUSE

Thank You