CYBAGE
Delivering Value. Scientifically.

GitLab

Presented by: Samiksha Deshpande

Agenda

- Introduction to GitLab?
- Why GitLab?
- GitLab Solutions
- GitLab Features
- GitLab Workflow
  - GitLab CI pipeline, GitLab Runner, Docker Executor
  - GitLab CI jobs and GitLab CI YAML file
- Comparison
  - GitLab CI vs GitHub Actions
  - GitLab vs Jenkins
- GitLab Integrations
- GitLab Security Configuration
- GitLab Best Practices
- GitLab Pricing
- Demo

Introduction to GitLab

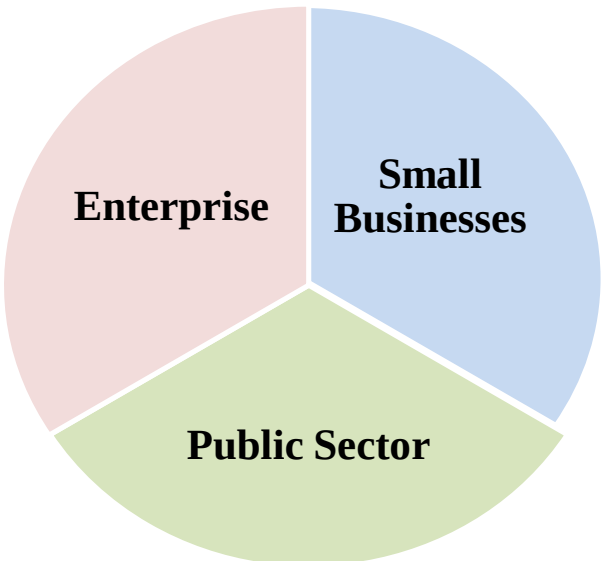
- GitLab is a web-based DevOps cycle tool
- It provided Git Repo Manager, wiki, issue tracking, CI/CD pipeline
- It was created by Ukrainian developer Dmytro Zaporozhets
- Its an Open-Source license develop by GitLab Inc.
- It is being developed by 171 team members in 31 different countries
- More than 1200 people contributed in total
- Allows all repositories to be up to 10 gigabytes in size
- Runs on Ruby on Rails platform
- Some companies using GitLab: *Sony, IBM, NASA, Oracle, Alibaba.*

Why GitLab?

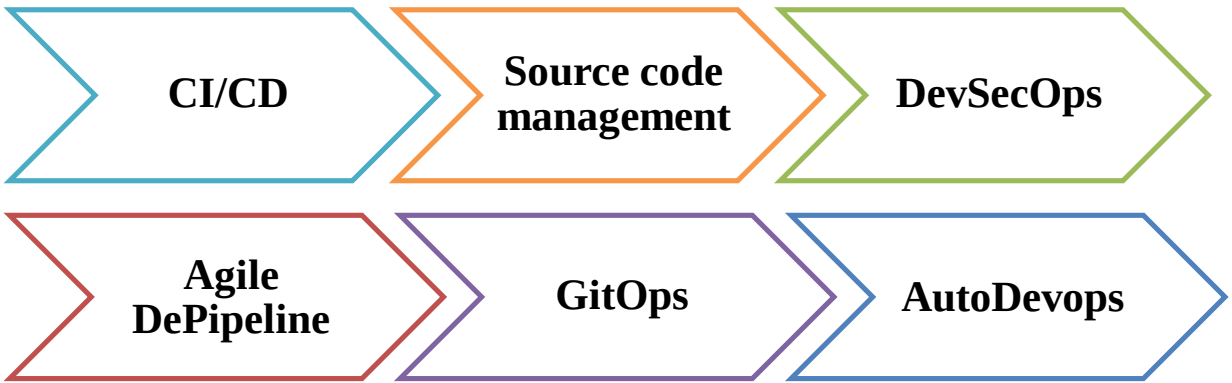
- 7x Faster cycle time
- Integrated AI across the software development lifecycle
- Privacy-first AI, with the right model for the right use case
- Security automation and governance at every step
- End-to-end compliance and auditability
- Flexible deployment
- End-to-end metrics and visibility across the software delivery lifecycle
- Built-in Enterprise Agile Delivery
- A multi-cloud strategy with no vendor lock-in
- An open core DevSecOps platform that you can make your own

GitLab Solution

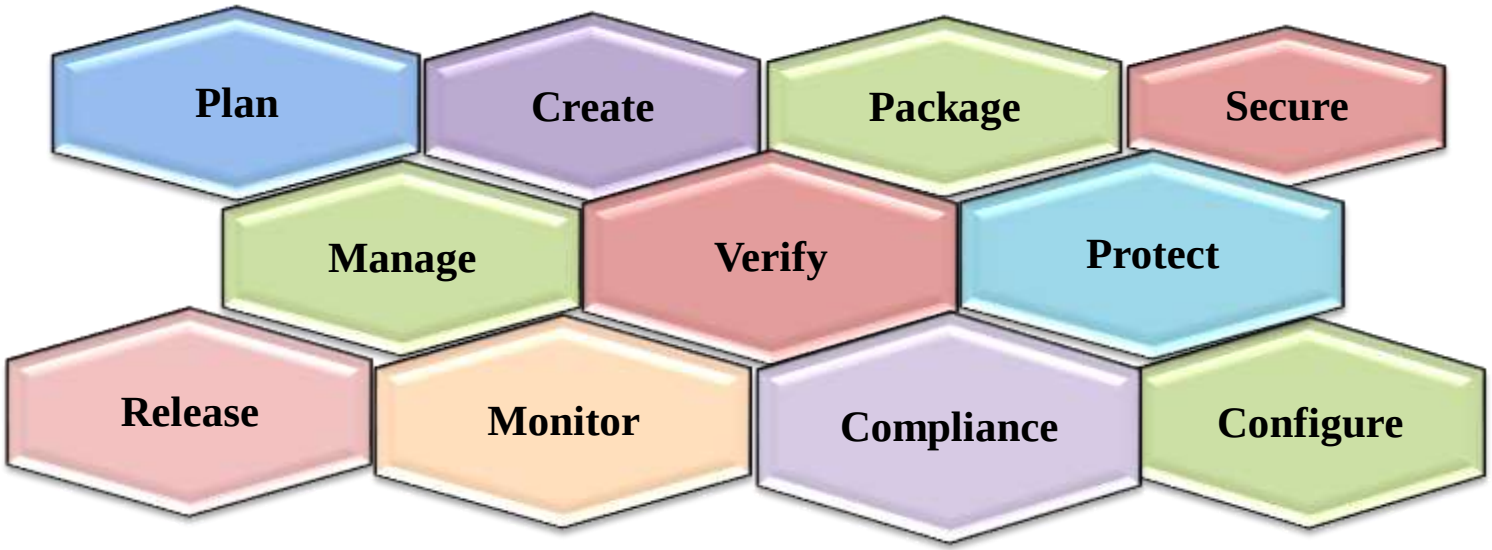
By organization type



By Use Cases



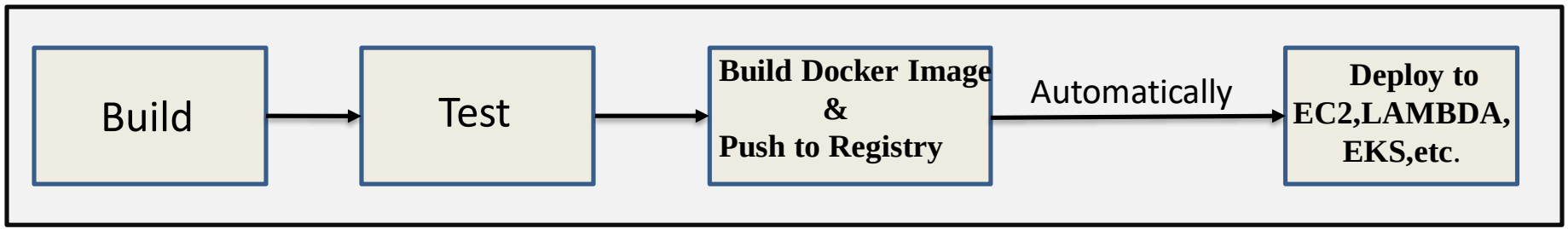
GitLab Features



GitLab Workflow

1) GitLab CI Pipeline

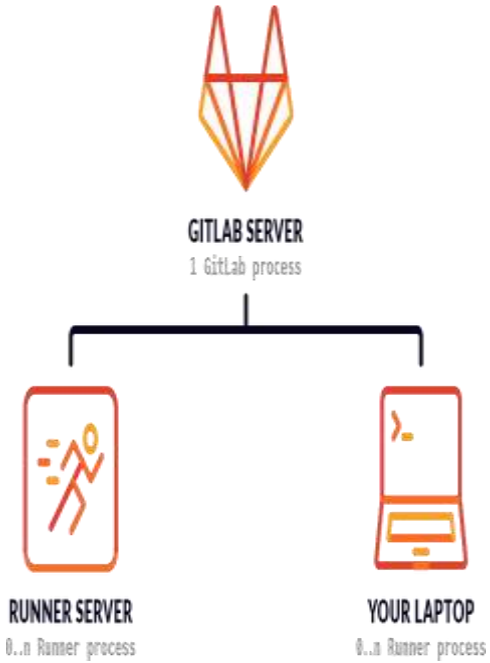
- A GitLab pipeline is a series of jobs used to build, test and deploy code, whenever a new commit is pushed to a GitLab repository



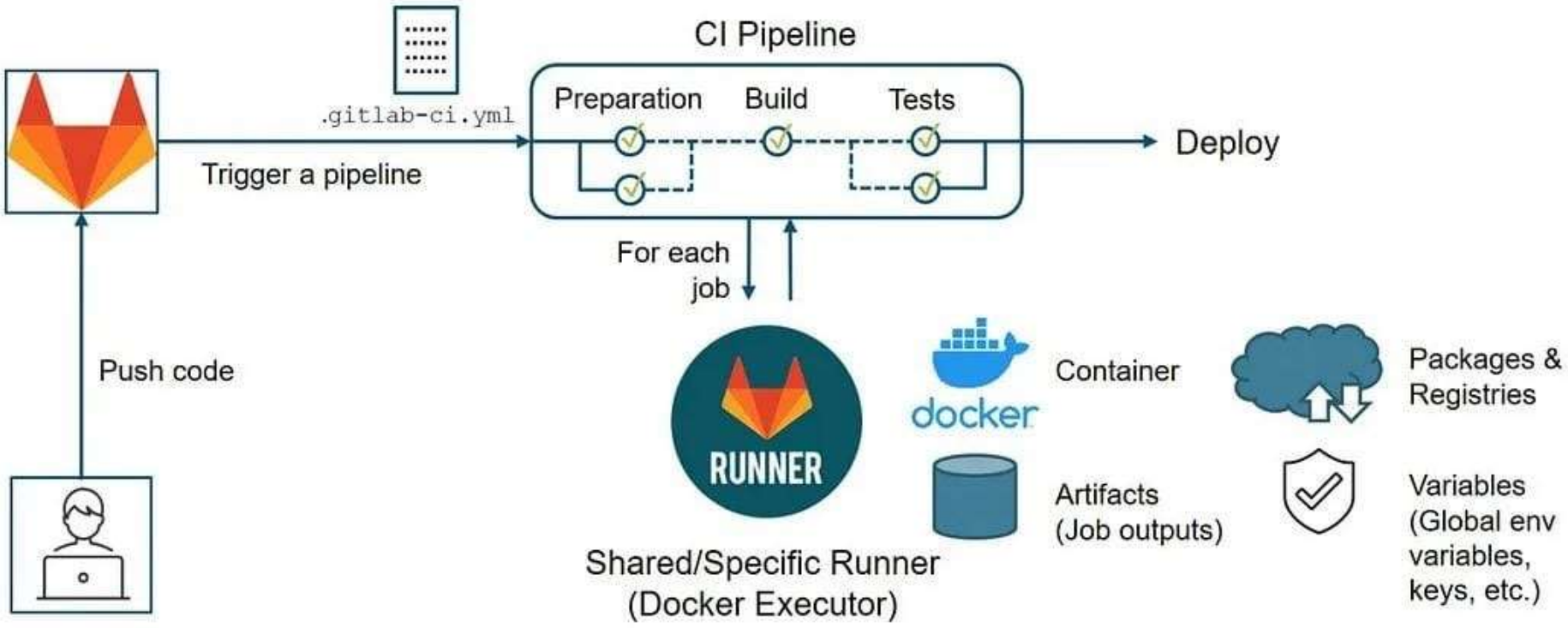
2) GitLab Runners

- They execute GitLab CI Jobs
- They can be installed on Windows or Linux system as a service
- Runners must be registered with GitLab as a shared runner or locked runner to a specific project/group

- The runner is an example of an assembly that can be used to run jobs on multiple machines.
- It can be installed on GNU/ Linux, macOS, FreeBSD, and Windows
- Types of Runners:
    - **GitLab-hosted runners/Instance Runner/Shared Runner**
    - **Self-managed runners/Project Runner/Specific Runner**
- It can be used in 5 types Shared, Group, Specific Runners, Locked, and Paused
    - **Shared:** All of your projects are available for CI/CD tasks.
    - **Group:** It can be used for all projects within the team or group you have defined.
    - **Specific:** It only works on the defined repository.
    - **Locked:** Runner cannot be assigned to other projects.
    - **Paused:** Runner can't get a new job and process it.



GitLab Diagram:



4) Choosing a Runner

- Shared : Registered to GitLab group and can be used by any project in that group
- Dedicated : Registered to a specific project

5) Cl Jobs

- A job is a series of commands that runs on the GitLab Runner
- Each Job Generally fulfils one "task" in the process
- The output from each job is collected and logged in the GitLab pipeline logs

6) .gitlab-ci.yml file

- The YAML file that defines a series of Variables, Stages, Jobs that get executed with every new pipeline
- A basic job consists of these elements:
  - Stage : Determines order in which the jobs are executed
  - Script : Commands that are executed by the runner

Executors:

- GitLab Runner implements a number of executors that can be used to run your builds in different environments

- GitLab Runner provides the following executors:

  - SSH
  - Shell –(commands will directly execute on runner machine)
  - Parallels
  - Virtual Box
  - Docker
  - Kubernetes
  - Custom

Selecting the executor :

- The executors support different platforms and methodologies for building a project.
- The table below shows the key facts for each executor which will help you decide which executor to use.

Executor	SSH	Shell	VirtualBox	Parallels	Docker	Kubernetes	Custom
Clean build environment for every build	✗	✗	✓	✓	✓	✓	conditional (4)
Reuse previous clone if it exists	✓	✓	✗	✗	✓	✗	conditional (4)
Runner file system access protected (5)	✓	✗	✓	✓	✓	✓	conditional
Migrate runner machine	✗	✗	partial	partial	✓	✓	✓
Zero-configuration support for concurrent builds	✗	✗ (1)	✓	✓	✓	✓	conditional (4)
Complicated build environments	✗	✗ (2)	✓ (3)	✓ (3)	✓	✓	✓
Debugging build problems	easy	easy	hard	hard	medium	medium	medium

Compatibility chart

Supported features by different executors:

Executor	SSH	Shell	VirtualBox	Parallels	Docker	Kubernetes	Custom
Secure Variables	✓	✓	✓	✓	✓	✓	✓
.gitlab-ci.yml: image	✗	✗	✓ (1)	✓ (1)	✓	✓	✓ (via $CUSTOM_ENV_CI_JOB_IMAGE)
.gitlab-ci.yml: services	✗	✗	✗	✗	✓	✓	✓
.gitlab-ci.yml: cache	✓	✓	✓	✓	✓	✓	✓
.gitlab-ci.yml: artifacts	✓	✓	✓	✓	✓	✓	✓
Passing artifacts between stages	✓	✓	✓	✓	✓	✓	✓
Use GitLab Container Registry private images	n/a	n/a	n/a	n/a	✓	✓	n/a
Interactive Web terminal	✗	✓ (UNIX)	✗	✗	✓	✓	✗

Docker Executor:

- GitLab Runner uses the Docker executor to run jobs on Docker images.
- It has Pre - build images
- We can execute this on top of existing Docker image

You can use the Docker executor to:

- Maintain the same build environment for each job.
- Use the same image to test commands locally without the requirement of running a job in the CI server.

- The Docker executor uses Docker Engine to run each job in a separate and isolated container. To connect to Docker Engine, the executor uses:
  - The image and services you define in .gitlab-ci.yml.
  - The configurations you define in config.toml.

Example :

image: maven:3.8.5-openjdk-17

Workflow (continued)

- **Some jobs may require additional components**
  - Dependencies : Job won't run until a specified prior job succeeds
  - Only : job only run when certain refs (e.g. branches or tags) change
  - When : set job to run based on result of prior jobs or by manual trigger
  - Tags : job will only run on runners that are registered with the specified tag
  - Artifacts : upload specific build artifacts to GitLab after job
  - Environment : tags the name and URL of the environment that your job is running on

- **Each job will run is highly customizable, jobs can trigger based on:**
  - Changes to specific branch
  - Variable comparisons changes to specific files

.gitlab-ci.yml file (Example)

```
stages:
  - unittest
# CI STEPS: Unit testing
unittest:
 stage: unittest
image: maven:3.8.5-openjdk-17
#when: manual
allow_failure: true
tags:
        - myrunner
script:
        - echo "Executing SpringBoot scenarios with maven"
        - mvn test
```

(Refer the dashboard)

Comparison : GitLab CI vs GitHub Actions

Parameters	GitHub Actions	GitLab CI
Product Type	Self-hosted/On-premise	Self-hosted/On-premise
Marketplace	Yes	No
Kubernetes Monitoring & Management	No, Build in integration	Yes, Build in integration
Distinct Native Security Scanning	No in build support	SAST, DAST, Fuzz-testing, Secret Scanning, Dependency Scanning, Container Scanning, License Compliance and vulnerability management all in one for a single cost
Retry Job	No, option are available	We can retry specific job

GitLab vs Jenkins

Parameters	GitLab CI	Jenkins
Language	Ruby	Java
Open-source or commercial	Open-Source	Open-source
Product Type	Self-hosted/On-Premise	Self-hosted/On-Premise
Built-in CI CD	Has both as a built-in feature	Depending on requirement we need to install plugin
Kubernetes Monitoring and Management	Build in integration	No, Build in integration
Distinct Native Security Scanning	SAST, DAST, Fuzz-testing, Secret Scanning, Dependency Scanning, Container Scanning, License Compliance and vulnerability management all in one for a single cost	No native support for security scanning, Requires integrating 3rd party plugins resulting in increased risk, added cost and technical support gaps
Built in Container Registry	Yes	No
Code Quality	It Provides ready templates checking the quality of code that you can include in yaml file	No ready template available for checking the code quality

GitLab Integrations

There are many integrations with numerous tools some are as fallows :

- GitLab has integration available for other CI/CD tools such as Atlassian Bamboo, Build kite, Jenkins, Jet Brains TeamCity
- Monitor you application health with Prometheus and dashboard as GitLab has an integration with Prometheus
- It can be integrated with Datadog which will send CI/CD pipeline information to Datadog to monitor for job failures and troubleshoot performance issue
- Jira is also integrated as project's issue tracker
- To get notified about your work GitLab has integration with Google chat, Microsoft teams, Slack notification

GitLab Security Configuration

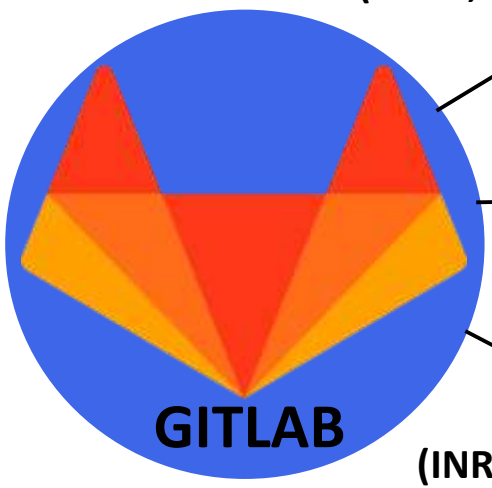
GitLab Security Conf.	Tier	Offering
Static Application Security Testing (SAST)	Free, Premium, Ultimate	GitLab.com, Self-managed, GitLab Dedicated
Infrastructure as Code (IaC) Scanning	Free, Premium, Ultimate	GitLab.com, Self-managed, GitLab Dedicated
Dynamic Application Security Testing (DAST)	Ultimate	GitLab.com, Self-managed, GitLab Dedicated
Dependency Scanning	Ultimate	GitLab.com, Self-managed, GitLab Dedicated

GitLab Security	Tier	Offering
Container Scanning	Free, Premium, Ultimate	GitLab.com, Self-managed, GitLab Dedicated
Application Security	Ultimate	GitLab.com, Self-managed, GitLab Dedicated
Pipeline Secret Detection	Free, Premium, Ultimate	GitLab.com, Self-managed, GitLab Dedicated
Coverage Fuzzing	Ultimate	GitLab.com, Self-managed, GitLab Dedicated
API Fuzzing	Ultimate	GitLab.com, Self-managed, GitLab Dedicated

GitLab Best Practices

- Use feature branches rather than direct commits on the main branch
- Test all commits, not only ones on the main branch
- Run every test on all commits
- Perform code reviews before merging into the main branch
- Deployments are automatic based on branches or tags
- Tags are set by the user, not by CI
- Releases are based on tags
- Fix bugs in main first and release branches second
- Commit messages reflect intent

GitLab Pricing



$29 (INR:2,414.52)

Premium

- Code Ownership and Protected Branches
- Code Quality Reports
- Enterprise User and Incident Management
- Merge Requests with Approval Rules
- 100GB transfer per month
- 10,000 compute credits per month

Free

- 5GiB storage
- 10GiB transfer per month
- 400 compute minutes per month
- 5 users per top-level group

$99 (INR:8,242.67)

Ultimate

- Dynamic Application Security Testing
- Security Dashboards
- Vulnerability Management
- Dependency Scanning
- Container Scanning
- 500GB transfer per month
- 50,000 compute credits per month

References:

1. https://docs.gitlab.com/
2. https://docs.gitlab.com/runner/
3. https://docs.gitlab.com/ee/ci/pipelines/
4. cybage-devops/java-application--gitlab: java-cicd-application (github.com)

CYBAGE
Delivering Value. Scientifically.

Thank You!