Preface –

1. What is Version Control System (VCS)?
2. What is Distributed Version Control System?
3. What is CI/CD?
4. Introduction to Git, GitBash and Gitlab
5. GitHub Vs GitLab
6. Local Vs Online
7. Gitlab Signup
8. Installation of Git Bash in Windows
9. Installation of Git Bash in Linux
10. Overview of Gitlab
11. What is GitLab Project?
12. How to create a GitLab Project?
13. What is GitLab Runner?
14. https://about.gitlab.com/pricing/

Just FYI on Differentiation: -

Git

- Git is a Version Control System
- it is a Hosting Service
- Supports Branching
- Can help with easy Merging

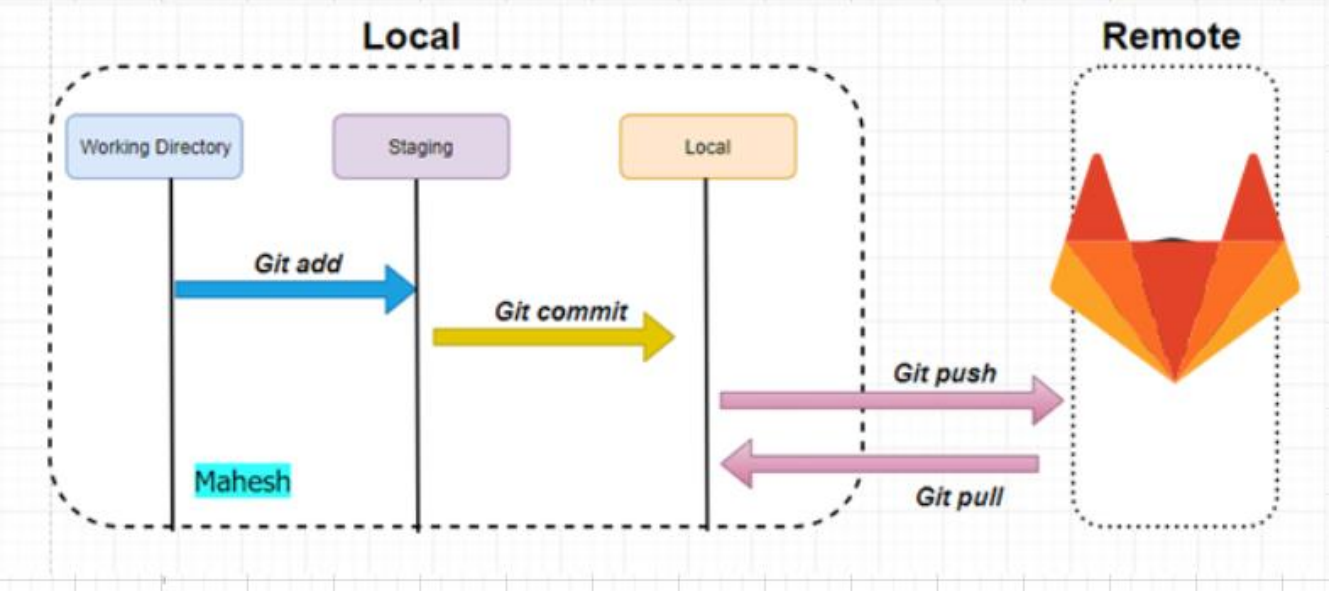
GitHub

- Online Repo to store your data
- Graphical web interface
- Acquired by Microsoft
- Private Repositories will be chargeable

GitLab

- Online Repo to store your data
- Graphical web interface
- Open Source
- Continuous Integration (CI)
- Continuous Delivery (CD)
- Issue Tracker
- Manage Whole DevOps Lifecycle
- Public and Private both will be free cost
- CI/CD Integration will be chargeable
- SDLC

Local Vs Remote (online)

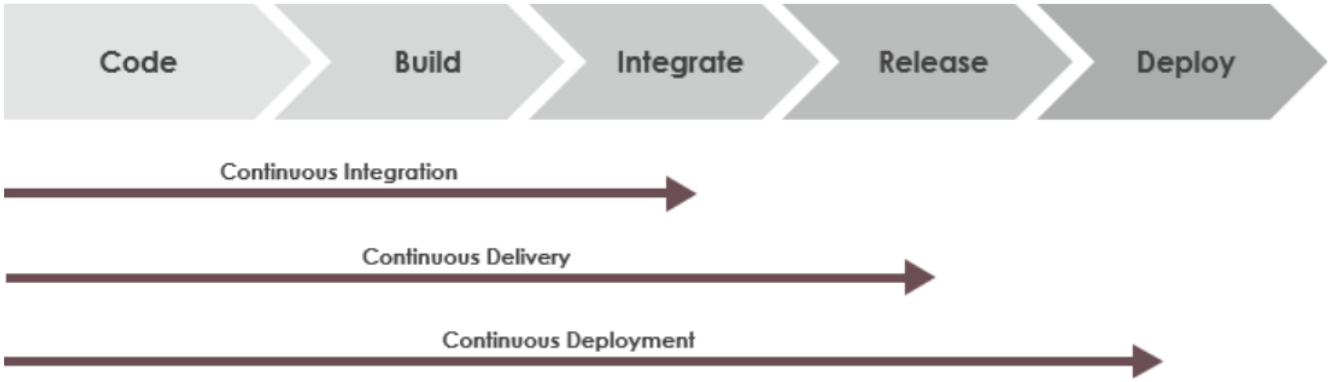


Let's begin with scenario-oriented labs: -

15. Introduction to Git Commands
16. Download Project from online to local (Clone)
17. Push Project from Local to online (Git Push)
18. GitLab Branching
19. Gitlab Groups/Members
20. GitHub Fork and GitLab Fork equal? (Fork)
21. Can I able to push code from GitLab to GitHub? (Mirroring)
22. Create SSH Keys
    a. Why do we need SSH keys: https://docs.gitlab.com/ee/ci/ssh_keys/
    b. Add SSH Keys to your GitLab Account
23. GitHub Vs GitLab

VCS	GitHub	Gitlab
Code, Git Container, Project Specific Data Code Discussions	Git Repository	Project
Merge Feature Branch into Master Branch	Pull Request	Merge Request
To Allow Group Level permissions, Users can add to Groups	Organizations/Collaborators	Groups/Members
CI/CD	None	Git Runner
Public Repository	Free of Cost	Free of Cost
Private Repository	Chargeable	Free of Cost
Fork	Online to Online	Within Project

CI/CD



GitLab Runner:

GitLab Runner is written in "GoLang" Open source Programming language that makes it easy to build simple, reliable, and efficient software.

Though it written in GoLang, there is no language specific to design your code.

GitLab Runner can be –

- Install using GitLab's repository for Debian/Ubuntu/CentOS/RedHat (preferred).
- Install on GNU/Linux manually (advanced).
- Install on macOS.
- Install on Windows.
- Install as a Docker service.
- Install in autoscaling mode using Docker machine.
- Install on FreeBSD.
- Install on Kubernetes.
- Install the nightly binary manually (development).

Git Lab Configurations

1. Installation
2. Register
3. Start/stop
4. Activation check

Installation of GitLab Runner in Windows Machine: -

- Login to your windows machine as a administrator
- Go to any location of your server or choose C:\ Drive and create a folder like “C:\Gitlab-Runner”
- Download GitLab Runner from this location referring your OS type
    - https://gitlab-runner-downloads.s3.amazonaws.com/latest/binaries/gitlab-runner-windows-386.exe
    - https://gitlab-runner-downloads.s3.amazonaws.com/latest/binaries/gitlab-runner-windows-amd64.exe
- Rename that downloaded binary to “gitlab-runner.exe”
- Open your CMD Prompt as a Administrator
- Navigate to “C:\Gitlab-Runner” and type below commands:
    - To Install GitLab Runner :: gitlab-runner.exe install
    - To Start GitLab Runner:: gitlab-runner.exe start
    - To Check GitLab Runner is installed:: gitlab-runner - - version

Register GitLab Runner in Windows Machine: -

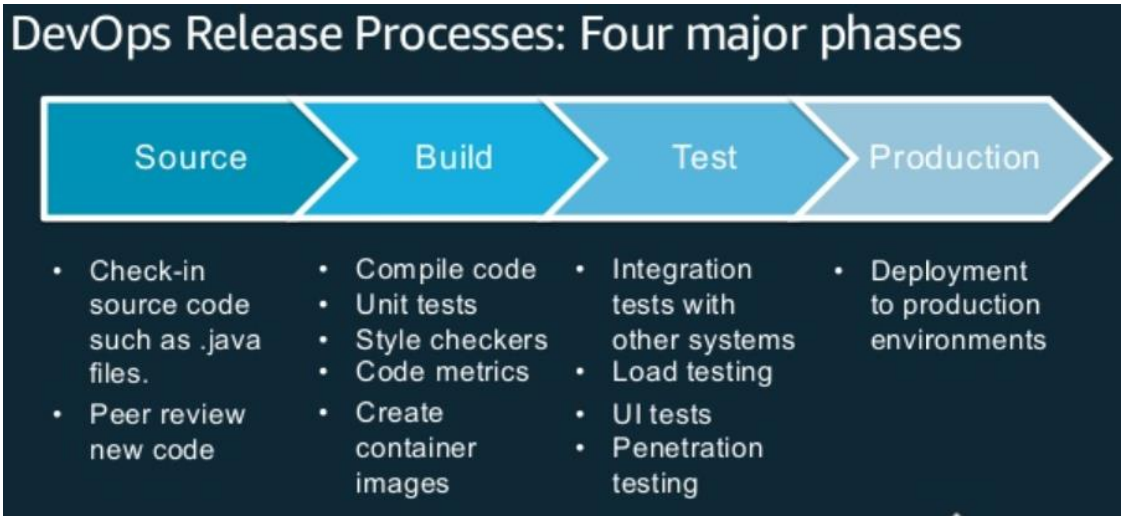
***Before registration don’t forget to stop the runner if it is already running

- To Stop GitLab Runner:: gitlab-runner.exe stop
- To start registration, run this command
    - gitlab-runner.exe register
- Please enter the gitlab-ci coordinator URL (e.g. https://gitlab.com)
    - https://gitlab.com
- Please enter the gitlab-ci token for this runner
    - You can find this in your GitLab project under CI/CD → Runners → “Setup a Runner Manually”.
- Please enter the gitlab-ci description for this runner
    - my-runner
- Please enter the gitlab-ci tags for this runner (comma separated):
    - CICD
- Please enter the executor: ssh, docker+machine, docker-ssh+machine, kubernetes, docker, parallels, virtualbox, docker-ssh, shell:
    - Docker (if its already installed) otherwise shell
- Runner Registered successfully, feel free to start it.
- Now you can restart your runner
    - To Start GitLab Runner:: gitlab-runner.exe start

Git Runner Activation at GitLab:-

- You can find Git Runner Activation in your GitLab project under CI/CD → Runners → “Runners Activated for this Project”.

Get started with GitLab Pipeline



Create a new file under your project and call that as "**.gitlab-ci.yml**"

<Reference URL: https://gitlab.com/help/ci/yaml/README>

Then Commit your code and verify that code is validated by GitLab or not.

Follows you can see that Pipeline is running automatically.

Each stages of your pipeline will be completed if the code is valid.