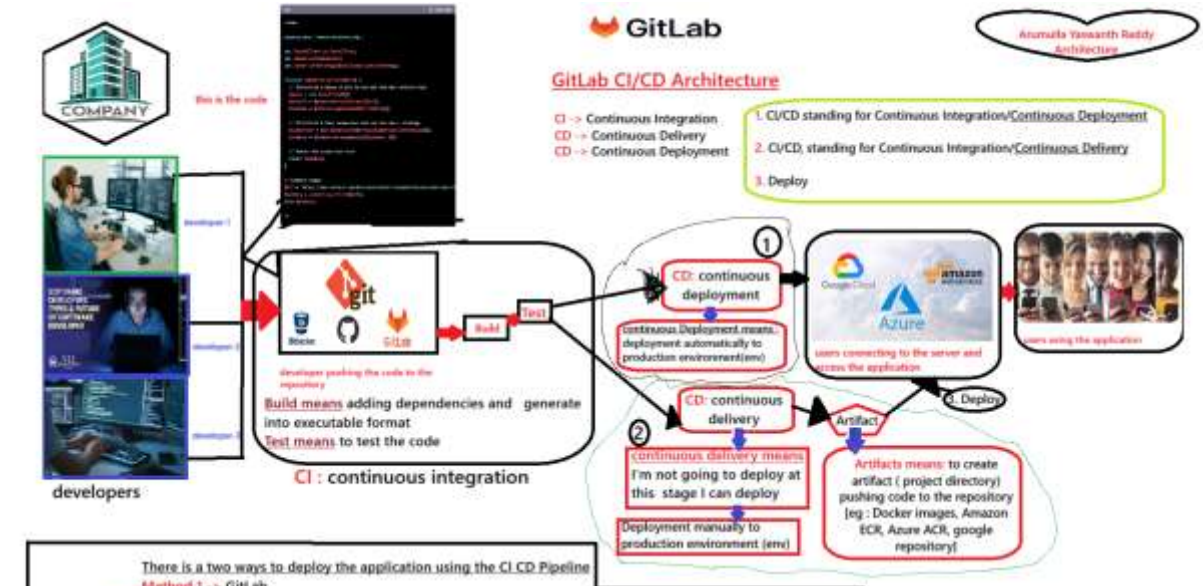
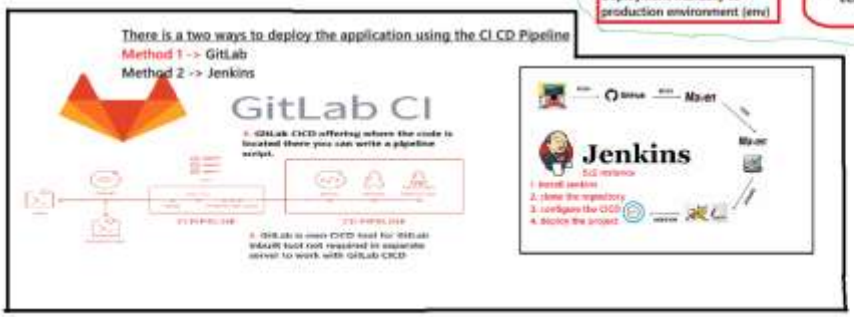

GITLAB CICD
CI/CD


GitLab
Anumala Yaswanth Reddy
Architecture
GitLab CI/CD Architecture
CI -> Continuous Integration
CD -> Continuous Delivery
CD -> Continuous Deployment
1. CI/CD standing for Continuous Integration/Continuous Deployment
2. CI/CD, standing for Continuous Integration/Continuous Delivery
3. Deploy
COMPANY
this is the code
developers
Build means adding dependencies and generate into executable format
Test means to test the code
CI : continuous integration
CD: continuous deployment
Continuous Deployment means deployment automatically to production environment(env)
CD: continuous delivery
Continuous delivery means I'm not going to deploy at this stage I can deploy
Deployment manually to production environment (env)
Artifact
Artifacts means: to create artifact (project directory) pushing code to the repository [eg : Docker images, Amazon ECR, Azure ACR, google repository]
Azure
users connecting to the server and access the application
users using the application
3. Deploy

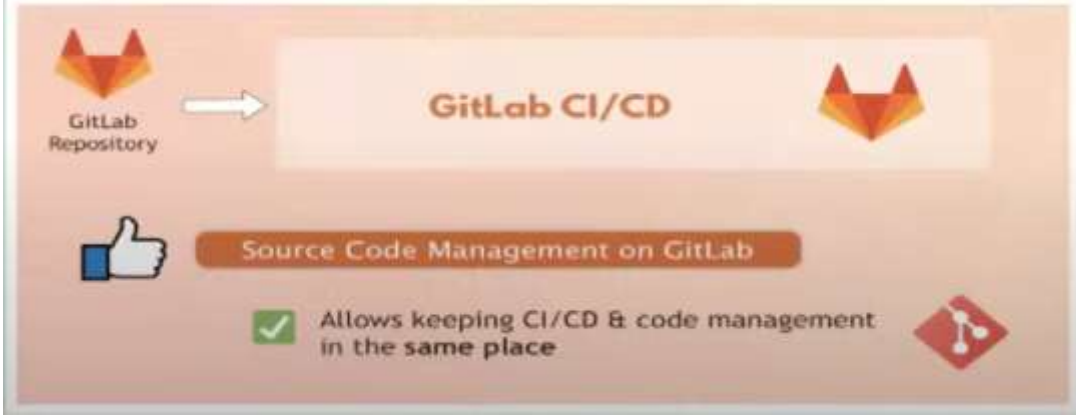

There is a two ways to deploy the application using the CI CD Pipeline
Method 1 -> GitLab
Method 2 -> Jenkins
GitLab CI
Jenkins
Maven

FEATURE	GITLAB	JENKINS
Version Control	Git	N/A (requires integration with a separate VCS tool).
Continuous Integration	Yes, built-in.	Yes, built-in.
Continuous Delivery	Yes, built-in.	Requires plugins or scripting.
Security	Built-in security features.	Requires plugins or scripting.
Code Review	Built-in code review features.	Requires plugins or scripting.
Performance	Generally faster due to built-in Git repository	May require additional resources for performance
Scalability	Scales well for small to medium-sized teams.	Scales well for large teams.
Cost	Free for self-hosted and cloud-hosted versions.	Free for self-hosted and has a cost for cloud-hosted.
Community	Active open-source community and enterprise support.	Active open-source community and enterprise

Many CI/CD platforms

Of course there are many CI/CD tools, one of the most used ones in the industry still being Jenkins. And Gitlab CI/CD is just one of those other CI/CD platforms

All of them have their advantages and disadvantages. But a big advantage of using Gitlab to build CI/CD pipelines for your applications is that you already have your code on Gitlab, so this is an extension of your software development processes in your team, where you can also build CI/CD pipelines on the same platform:



How do we create a Gitlab CI/CD pipeline?

In the **root of the project's repository**, we're going to create this YAML file and we're going to write all the pipeline configuration inside this configuration has to be called **.gitlab-ci.yml**