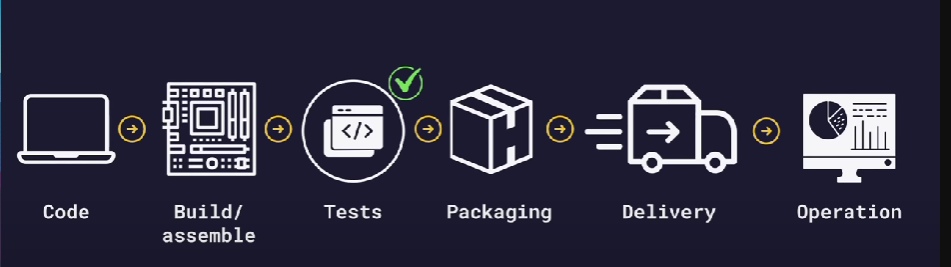
Gitlab

Pipeline



A DevOps pipeline is a set of automated processes and tools that allows developers and operations professionals to collaborate on building and deploying code to a production environment.

YAML :

Superset of JSON

JSON easier to parse and YAML easier to read and little hard to write

Used to store configuration values and it uses key value pair

Basic syntax :

Use colon to separate keys and values (for values need a space between colon and value)

Use indentation for separation of values

For list we use (-) before the values

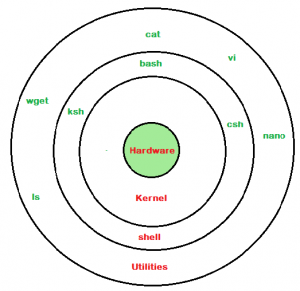
A screenshot of a computer program

Description automatically generated

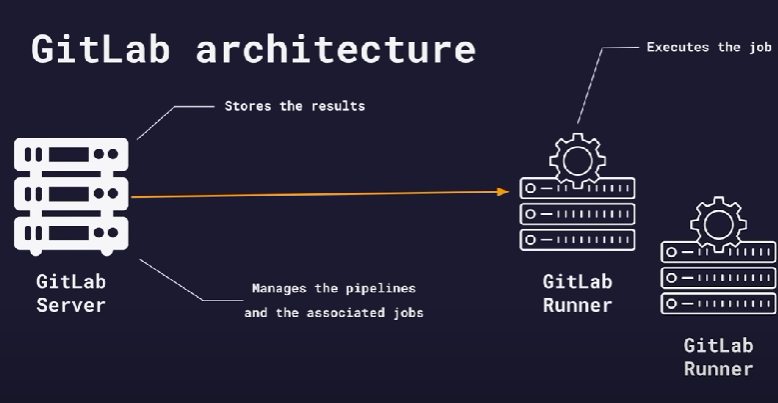
Shell :

Shell is the command line interface of the system used from communication with the system

A shell is a special user program that provides an interface for the user to use operating system services. Shell accepts human-readable commands from users and converts them into something which the kernel can understand.



Gitlab architecture



A screenshot of a computer

Description automatically generated

Pipeline stages:

E.g. : we need to test after the build

A computer screen shot of a program

Description automatically generated

Exit code => 0 => success

Exit code => 1 - 255 => error

The above test job is failing because the build computer job after completes the tasks it will delete the containers and folders created in it.

To pass the files to other jobs we need to store the important output files of the job to the coordinator(server) by sending the output by artifacts

A screenshot of a computer program

Description automatically generated

Testing the build :

Use grep command for checking specific strings in files

A computer screen with text

Description automatically generated

Variables:

Variables are used in yaml to avoid repetition

We can use variables in three ways:-

* Globally
* Inside variables in a specific job
* Inside scripts of specific job

Use quotes for values

Prefer using uppercase with \_

Access variables by using $

Inside scripts don’t use “”

A screenshot of a computer program

Description automatically generated

DevOps

DevOps is a combination of software development (dev) and operations (ops). It is defined as a software engineering methodology which aims to integrate the work of development teams and operations teams by facilitating a culture of collaboration and shared responsibility.

Stemming from an [Agile approach](https://about.gitlab.com/topics/agile-delivery/) to software development, a DevOps process expands on the cross-functional approach of building and shipping applications in a faster and more iterative manner.

**Core DevOps principles**

Automation of the software development lifecycle.

Collaboration and communication

Continuous improvement and minimization of waste.

Hyperfocus on user needs with short feedback loops.

CI/CD

Continuous integration (CI) is practice that involves developers making small changes and checks to their code. Due to the scale of requirements and the number of steps involved, this process is automated to ensure that teams can build, test, and package their applications in a reliable and repeatable way. [CI](https://www.synopsys.com/glossary/what-is-continuous-integration.html) helps streamline code changes, thereby increasing time for developers to make changes and contribute to improved software.

[Continuous delivery](https://www.synopsys.com/glossary/what-is-continuous-delivery.html) (CD) is the automated delivery of completed code to environments like testing and development. CD provides an automated and consistent way for code to be delivered to these environments.

[Continuous deployment](https://www.synopsys.com/glossary/what-is-continuous-development.html) is the next step of continuous delivery. Every change that passes the automated tests is automatically placed in production, resulting in many production deployments.

….CI is a set of practices performed as developers are writing code, and CD is a set of practices performed after the code is completed….

Building the project



Disable job

Add . before job

Deploy in aws

Create account in aws

Create a s3 bucket

Use aws cli for interacting with aws cloud services

Use aws cli image from docker

Uploading files to aws

First create s3 bucket in aws

Use following commands to upload file

echo "hello world" > text.txt

aws s3 cp text.txt s3://abinesh-24102002-1

variables

