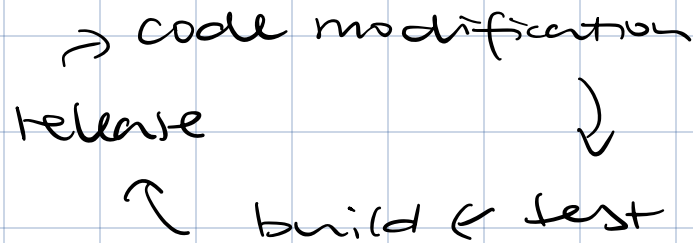


## · CI/CD

continuous integration / deployment



## · GitLab Architecture

- **GitLab Instance/Server** (SaaS, maintained by GitLab)
  - Host application code and pipeline config
  - Manage pipeline exe
- **GitLab Runner**
  - Run CI/CD projects
  - Get assignment from Server

## · Demo (Python)

- Executing test is a core part of CI/CD pipeline
  - verify new code changes, don't break anything.
  - if test fails, then pipeline fails, new change won't deploy
- pipeline config file

pipeline is scripted, hosted inside application's git repository

→ whole CI/CD config is written in YAML format.

and file name is .gitlab-ci.yml

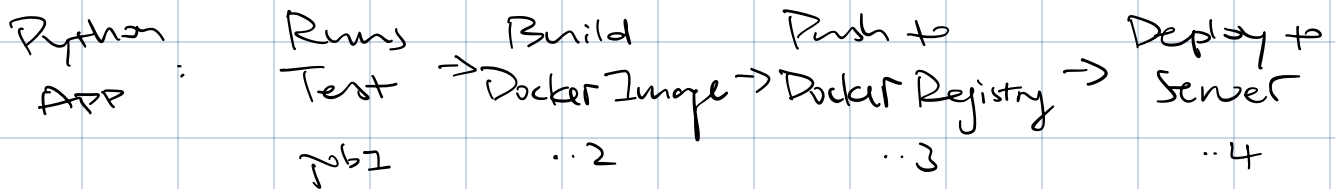
## - Types of Executor

- local Windows / Linux / MacOS ..
- Docker container.

- only Docker itself needs to be installed
- each job runs in a separate & isolated container
- container run based on Docker Image.

more like a  
blueprint

a read-only snapshot that contains  
everything needed to run



## - Build & Push Docker Image

- Docker Hub
- Don't hardcode

Settings → CI/CD → Variable

```
1 variables:
2   IMAGE_NAME: nanajanashia/demo-app
3   IMAGE_TAG: python-app-1.0
4
5
6 run_tests:
7   image: python:3.9-slim-buster
8   before_script:
9     - apt-get update && apt-get install make
10  script:
11    - make test
12
13
```

Pipeline level  
variable

```
build_image:
  image: docker:20.10.16
  services:
    - docker:20.10.16-dind
  variables:
    DOCKER_TLS_CERTDIR: "/certs"
  before_script:
    - docker login -u $REGISTRY_USER -p $REGISTRY_PASS
  script:
    - docker build -t $IMAGE_NAME:$IMAGE_TAG .
    - docker push $IMAGE_NAME:$IMAGE_TAG
```

```
build_image:
  before_script:
    - docker login -u $REGISTRY_USER -p $REGISTRY_PASS
  script:
    - docker build -t $IMAGE_NAME:$IMAGE_TAG .
    - docker push $IMAGE_NAME:$IMAGE_TAG
```

← compute version

Docker in Docker.

need docker

available inside

docker

- Stage

- Group/Define multiple jobs into stages.

to make the parallel or in order

```
5  stages:
6    - test
7    - build
8
9  run_tests:
10   stage: test
11   image: python:3.9-slim-buster
12   before_script:
13     - apt-get update && apt-get install make
14   script:
15     - make test
16
17   abc build
18   build_image:
19     stage: build
20     image: docker:20.10.16
21   services:
```