**Continuous Integration / Continuous Deployment (CI/CD)**

Below is the solution for Continuous Integration / Continuous Deployment of a web application using GITLAB. The advantages of GITLAB are, it’s an all-in-one platform with single interface, easy to use and it’s scalable.

Web application is assumed to be using html and python. Relevant tools are considered in the below pipeline stages.

**Defining the repository structure:**

The repository should have below branches.

1. Feature
2. Develop
3. Staging
4. Master

**Code check-in and merging strategies:**

* When developer checks-in the code to feature branch then following stages are performed:

*Unit Test -> Build -> Security Test -> Deploy Feature*

* When developer merges code from feature to develop branch following stages are performed:

*Deploy Development -> Load Test*

* When developer merges code from develop to staging branch following stages are performed:

*Deploy Staging -> Behavior Testing*

* When developer merges code from staging to master branch following stages are performed, there may be a manual step for approval to merge staging to master branch:

*Deploy Production*

**Environments:**

The pipeline stages are divided into 4 environments:

1. Feature
2. Development
3. Staging
4. Production

**The pipeline here should have below stages in the below sequence:**

Stages:

- Lint and Unit Test

- Build and Package

- Static Security Test

- Deploy Feature

- Deploy Develop

- Unit Test

- Load Test

- Deploy Staging

- Behavioral testing

- Deploy Production

1. Lint and Unit Test -

Tools: flake8 and Pytest

Description: flake8 is used here to test code linting and style checking. Pytest is used here for unit test for python code.

1. Build and Package -

Tools: Docker (Dockerfile)

Description: Create Dockerfile and build docker image. Push to GITLab container registery.

1. Static Security Test –

Tools: WhiteSource

Description: WhiteSource is used to test the vulnerabilities in the Docker image like, self signed certificates or untrusted packages.

1. Deploy Feature –

Tools: Docker

Description: Deploy the feature to Heroku- Feature environment. This is a separate environment where individual features are tested by developers. It is a lower level environment. This feature environment/branch is deleted when it is further merged to development environment.

1. Deploy Development –

Tools: Docker

Description: Deploy the feature to Heroku- Development environment. This is a separate environment.

1. Unit Test –

Tools: Selenium

Description: This is step is to test the code deployed in development environment. Multiple features are tested at once.

1. Load Test -

Tools: JMeter

Description: JMeter is used for load testing, performance testing, and functional testing of web application

1. Deploy Staging –

Tools: Docker

Description: Deploy the feature to Heroku- Staging environment. This is a separate environment with separate deployment url.

1. Behavioral Testing –

Tools: Cucumber and Xray

Description: Use Cucumber framework for behavior testing and use Xray for checking the test reports. Email is sent if there are any errors in the report. A manual approval process is introduced after behavioral test is completed.

1. Deploy Production –

Tools: Docker

Description: Deploy the feature to Heroku- Production environment. This is a separate environment with separate deployment url.