**Локальная настройка GITLAB + gitlab runner + docker registry для CICD**

Так как работа предполагается в локальном пользовании, сертификаты не используем.

Так же и для docker registry. Все по HTTP.

1. Создаем записи типа A – gitlab и registry

2. GITLAB устанавливается на дистрибутив ubuntu server 20.04

3. В  /etc/gitlab/gitlab.rb - external\_url 'http://наш\_домен.local'

4. Реконфигурируем наш гитлаб - gitlab-ctl reconfigure

5. Пароль админа смотрим - /etc/gitlab/initial\_root\_password

На данном этапе гитлаб работает.

6. Устанавливаем связку – docker, docker-compose

7. Для целей CICD настраиваем раннеры. **RUNNER**

**Создаем том для Docker**:

docker volume create gitlab-runner-config , где <gitlab-runner-config> – название тома

**Запускаем контейнер GitLab Runner**, используя только что созданный том:

docker run -d --name gitlab-runner --restart always \

-v /var/run/docker.sock:/var/run/docker.sock \

-v gitlab-runner-config:/etc/gitlab-runner \

gitlab/gitlab-runner:latest

В нашем проекте гитлаб в меню-Settings-CI/CD-Runners находим информацию для добавления нового раннера – адрес и токен. Копи-пастим

Заходим во внутрь контейнера раннера –

docker exec -it <наш\_раннер> gitlab-runner register , где gitlab-runner register – команда регистрации

В ответ копи-пастим адрес и токен из проекта. Далее описание – можно по дефолту,

По поводу тэгов – обязательно нужны – данной опцией можно управлять какие раннеры должны выполняться. Так как собираем контейнеры то ставим tag – docker.

Далее выбираем экзикьютеры – выбираем – docker. Спросит какой использовать контейнер

для сборок - docker:20.10.22-dind . Это образ докер в докере. Именно 20.10.22 т.к. последняя имеет проблемы.

Проверяем в меню нашего проекта появление нашего docker раннера.

В процессе сборки может появится ошибка типа – ERROR: during connect…..

**лечится** –

Смотрим подробную инфу о нашем созданном томе –

docker inspect <наш\_том>

Находим, что он находится - /var/lib/volumes/<наш\_том>/\_data

Меняем каталог на - /var/lib/volumes/<наш\_том>/\_data

Редактируем - config.toml

Находим наш раннер и в строке – Volumes - в квадратных скобках через запятую к записи рядом – “/cache” добавляем – “/var/run/docker.sock:/var/run/docker.sock”

8. Настраиваем для нашего проекта – Container Registry

Он будет держаться гитлабом. И авторизация то же гитлабом. Для этого в - /etc/gitlab/gitlab.rb

раскомитим и дабавим:

registry\_external\_url 'http://registry.<наш\_домен>:5050' - работаем по http

gitlab\_rails['registry\_enabled'] = true - включаем внутренний репозитарий гитлаба

\_\_\_\_\_\_\_\_\_не проверял\_\_\_\_\_\_\_\_\_\_

registry['registry\_http\_addr'] = "127.0.0.1:5000"

registry['debug\_addr'] = "localhost:5001"

\_\_\_\_\_\_\_не проверял\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Далее реконфигурируем наш гитлаб - gitlab-ctl reconfigure

Но это еще не все. Отключаем https в docker – для этого –

создаем или редактируем - /etc/docker/daemon.json с содержимым –

{

"insecure-registries" : ["registry.<наш\_домен>:5050"]

}

Сохраняем и перезапускаем docker.

Проверяем – docker login registry.<наш\_домен>:5050

Пароль и логин от нашего гитлаба. Login Succeeded

Теперь все должно быть настроено!!!!