
ŚCIĄGA LARAVEL

CI/CD z Dockerem na GitLabie

Stage'y test → build → deploy, buildx, registry GitLab, deploy
przez SSH

TL;DR

- **test** — PHP + Node na obrazie deweloperskim
- **build** — 3 obrazy przez `docker buildx` + `dind`
- **deploy** — SSH → skrypt na serwerze
- `needs:` tworzy graf zależności, `only: [main]` ogranicza deploy do gałęzi głównej

Szkielet .gitlab-ci.yml

```
image: docker:latest

stages: [test, build, deploy]

variables:
  DOCKER_BUILDKIT: 1
  DOCKER_HOST: tcp://docker:2375/
  DOCKER_TLS_CERTDIR: ""
  IMAGE_NAMESPACE: twoj/laravel-production
  REGISTRY: registry.gitlab.com

services:
  - docker:dind

before_script:
  - echo "$CI_REGISTRY_PASSWORD" | docker login \
    -u "$CI_REGISTRY_USER" --password-stdin "$CI_REGISTRY"
```

Stage test – Node

```
test_node:
  stage: test
  image: twoj/php-8.4-fpm:latest
  before_script: [ npm ci, npm run build ]
  script: [ npm run format, npm run types, npm run lint ]
  artifacts:
    paths: [ node_modules/, public/build/ ]
    expire_in: 1 hour
  only: [ main, merge_requests ]
```

Stage test – PHP

```
test_php:
  stage: test
  image: twoj/php-8.4-fpm:latest
  needs: [test_node]
  before_script:
    - git config --global --add safe.directory $(pwd)
    - composer install --no-progress --no-interaction --prefer-dist
  script:
    - cp .env.testing .env && php artisan key:generate
    - composer larastan
    - composer pint
    - php artisan test --env=testing
  variables:
    DB_CONNECTION: sqlite
    DB_DATABASE: ":memory:"
  only: [ main, merge_requests ]
```

Testy dostają własne usługi (SQLite `:memory:`) — bez współdzielenia stanu z innym jobem.

Stage build – buildx

```
build_php:
  stage: build
  image: docker:24.0.5
  needs: [test_php, build_node]
  script:
    - docker buildx create --use
    - |
      docker buildx build \
        --platform linux/amd64 \
        --file docker/php/Dockerfile \
        --tag $REGISTRY/$IMAGE_NAMESPACE/php:latest \
        --push \
        --build-context node=docker-image://$REGISTRY/$IMAGE_NAMESPACE/node:latest \
        .
  only: [ main ]
```

`build_nginx` analogicznie + `--build-arg HTTPASSWD_USER=$HTTPASSWD_USER` . Kolejne obrazy używają `node` jako build-context.

Stage deploy – SSH

```
deploy_production:
  stage: deploy
  image: alpine:3.19
  needs: [build_node, build_php, build_nginx]
  before_script:
    - apk add --no-cache openssh-client
    - mkdir -p ~/.ssh
    - echo "$SSH_KEY" | tr -d '\r' > ~/.ssh/id_rsa
    - chmod 600 ~/.ssh/id_rsa
    - ssh-keyscan -p $SSH_PORT $SSH_HOST >> ~/.ssh/known_hosts
  only: [ main ]
```

Stage deploy – transfer + trigger

script:

```
- cat "$ENV_FILE" > .env
- |
  { echo "REGISTRY=$REGISTRY"; echo "IMAGE_NAMESPACE=$IMAGE_NAMESPACE";
    echo "TAG=latest";
    echo "CI_REGISTRY_USER=$CI_REGISTRY_USER";
    echo "CI_REGISTRY_PASSWORD=$CI_REGISTRY_PASSWORD"; } >> .env
- scp -P $SSH_PORT docker-compose.production.yml \
  $SSH_USER@$SSH_HOST:~/laravel/docker-compose.yml
- scp -P $SSH_PORT .env deploy.sh $SSH_USER@$SSH_HOST:~/laravel/
- ssh -p $SSH_PORT $SSH_USER@$SSH_HOST \
  "cd ~/laravel && chmod +x deploy.sh && ./deploy.sh"
```

Uwagi bezpieczeństwa

`DOCKER_TLS_CERTDIR: ""` wyłącza TLS dla `dind` — OK w izolowanym runnerze CI, nie kopiuj wzorca gdzie indziej.

- Poświadczenia registry w serwerowym `.env` muszą mieć zawężony zakres (tylko pull).
- Klucz w `~/.ssh/id_rsa` — rozważ `ed25519` zamiast RSA.
- Osobny klucz deploy i użytkownik `deployer`.

Checklist produkcyjny

- Deploy tylko z `main` , nigdy z MR.
- Jeden artefakt promowany przez stage'y — nie przebudowuj obrazu w `deploy` .
- Migracje jako nazwany krok w `deploy.sh` z `--force` ; plan rollbacku.
- `known_hosts` przez `ssh-keyscan` , nie `StrictHostKeyChecking=no` .
- Weryfikacja po deployu (`/health` , `artisan about --only=cache`).

PODSUMOWANIE

Ściąga w skrócie

- 3 stage'y: test → build → deploy, spięte przez `needs`
- Testy: własne usługi, artefakty z `expire_in`
- Build: `buildx` + `node` jako build-context
- Deploy: `alpine` + `openssh-client` + `ssh-keyscan`
- `only: [main]` — deploy nie z MR
- Scoped token registry, `ed25519` , `deployer`

Przydało się?

Zapisz tę ściągę na później.

Pełny artykuł: dominik-dev.pl

Udostępnij • Skomentuj