
CERN

Docker Image Testing in GitLab CI

Thomas Løkkeborg - IT-DB-DAR
github.com/tholok97

1 May 2019

Outline

Outline - We Are Here

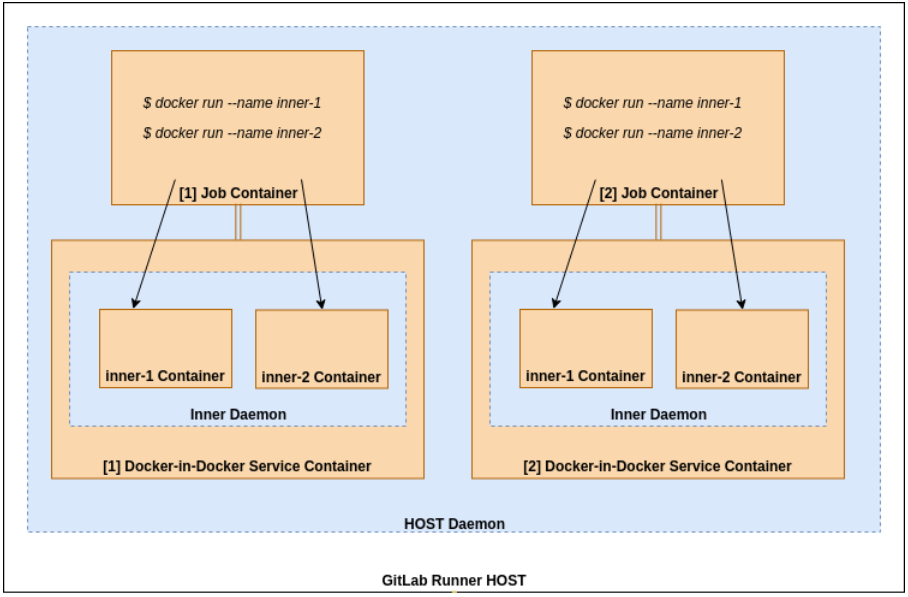
Features Used

- GitLab CI
- GitLab Container Registry
- Private GitLab Runners
- GitLab Runner Exec
- **Docker-in-Docker**
- **Artifacts**

GitLab

Features Used - Docker-in-Docker

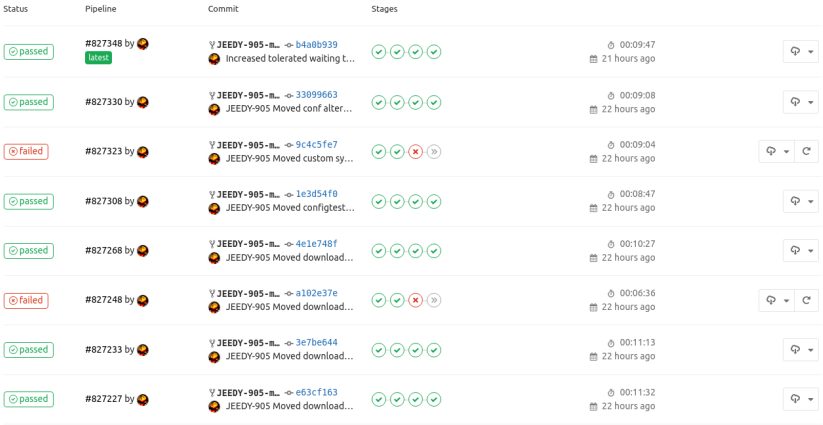
- Docker daemon running inside Docker daemon
- Access to `docker` command in GitLab CI jobs


$ docker run --name inner-1
$ docker run --name inner-2
[1] Job Container
$ docker run --name inner-1
$ docker run --name inner-2
[2] Job Container
inner-1 Container
inner-2 Container
Inner Daemon
[1] Docker-in-Docker Service Container
inner-1 Container
inner-2 Container
Inner Daemon
[2] Docker-in-Docker Service Container
HOST Daemon
GitLab Runner HOST

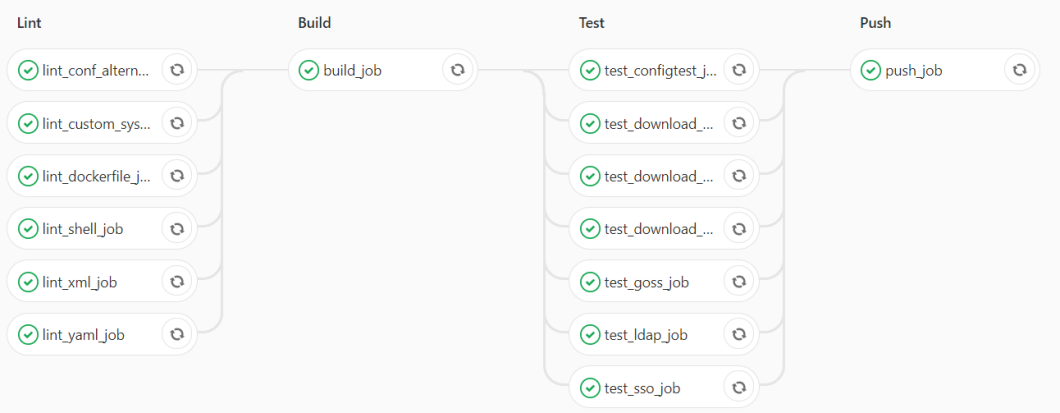
Features Used - Artifacts

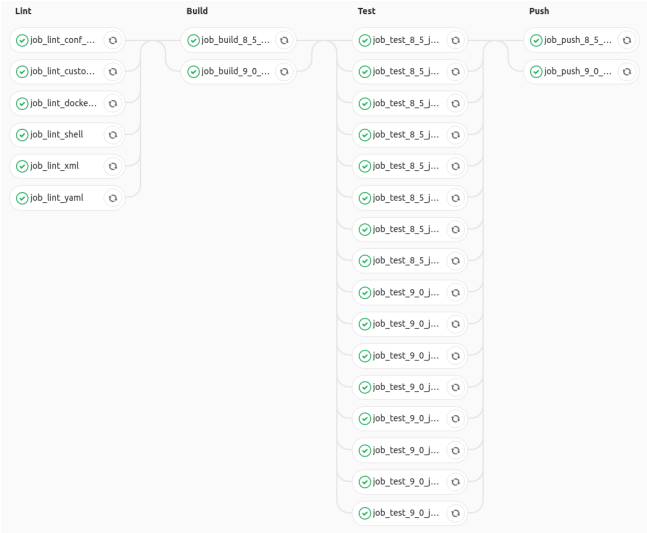
- Images passed between jobs as artifacts
- `docker save / load` - tarball

The Pipeline



The Pipeline*




Lint
job_lint_conf_...
job_lint_custo...
job_lint_docke...
job_lint_shell
job_lint_xml
job_lint_yaml
Build
job_build_8_5_...
job_build_9_0_...
Test
job_test_8_5_j...
job_test_8_5_j...
job_test_8_5_j...
job_test_8_5_j...
job_test_8_5_j...
job_test_8_5_j...
job_test_8_5_j...
job_test_8_5_j...
job_test_9_0_j...
job_test_9_0_j...
job_test_9_0_j...
job_test_9_0_j...
job_test_9_0_j...
job_test_9_0_j...
job_test_9_0_j...
job_test_9_0_j...
Push
job_push_8_5_...
job_push_9_0_...

The Pipeline - Quick facts

- Approx. duration of full pipeline: 13 minutes
- Jobs per pipeline: 26
- Total started pipelines: 362
- Success rate: 80%
- Biggest timestealer: Passing artifacts + `docker save/load`

Outline - We Are Here

Goss - Dgoss

- Asserts Docker image meets requirements described in goss.yaml file
- Tests:
  - `user`
  - `file`
  - `command`
  - `port`
  - `http`
  - `...`

Goss - goss.yaml

```
user:
  tomcat:
    uid: 1000
    gid: 1000
    exists: true
    groups:
      - tomcat
    shell: /sbin/nologin
http:
  http://localhost:8080/health-check/:
    status: 200
    no-follow-redirects: true
    timeout: 5000
    body:
      - I'm running
```

Goss - Running

```
dgoss run \
 -e SOME_VAR=some-value \
 -v /mount/path/outer:/mount/path/inner \
 image-under-test:1.0
```

Goss - Output

```
User: tomcat: exists: matches expectation: [true]
User: tomcat: uid: matches expectation: [1000]
User: tomcat: gid: matches expectation: [1000]
User: tomcat: home: matches expectation: ["/usr/apache-tomcat/tomcat8"]
User: tomcat: groups: matches expectation: [["tomcat"]]
User: tomcat: shell: matches expectation: ["/sbin/nologin"]
Command: grep "At least one JAR was scanned for TLDs yet contained no TLDs" /srv/tomcat/logs/catalina.log: exit-status: matches expectation: [1]
Command: grep -v  '^TOMCAT' /srv/tomcat/logs/catalina.log: exit-status: matches expectation: [1]
Command: grep "Server startup in" /srv/tomcat/logs/catalina.log: exit-status: matches expectation: [0]
Command: grep "Loaded APR based Apache Tomcat Native library" /srv/tomcat/logs/catalina.log: exit-status: matches expectation: [0]
HTTP: https://localhost:8443/health-check/: status: matches expectation: [200]
HTTP: https://localhost:8443/health-check/: Body: matches expectation: [I'm running]
HTTP: http://localhost:8080/health-check/: status: matches expectation: [200]
HTTP: http://localhost:8080/health-check/: Body: matches expectation: [I'm running]


Total Duration: 0.776s
Count: 47, Failed: 0, Skipped: 0
```

SSO Test - Concept

- «Given valid webapp and Identity Provider, can a test user log in successfully?»
- Keycloak as containerized Identity Provider with test data from JSON
- Webapp that shows received information about logged-in user
- Steps: Setup, Execute, Teardown

SSO Test - Setup

```
docker run \
 --name=sso-test-tomcat-container \
 -d \
 -p 8080:8080 \
 -e JEEDY_TOMCAT_SSO_IDP_LOGIN_BINDING_URL=\
    http://localhost:9090/auth/realms/testrealm/protocol/saml \
 -e JEEDY_TOMCAT_SSO_ENTITY_ID=sample-webapp-entity-id \
 -v "$PWD"/tests/sso/web.xml:/srv/tomcat/webapps/sample/WEB-INF/web.xml \
 test-image

docker run \
 --name=sso-test-keycloak-container \
 -d \
 -p 9090:8080 \
 -v "$PWD"/tests/sso/testrealm.json:/tmp/testrealm.json \
 -e KEYCLOAK_IMPORT=/tmp/testrealm.json \
 -e KEYCLOAK_USER=admin \
 -e KEYCLOAK_PASSWORD=admin \
 "${KEYCLOAK_IMAGE}"
```

SSO Test - Execute

Flow:

1. User tries to access web application
2. User is redirected to SSO login
3. User logs in
4. User is redirected back to web application
5. Assert that web application shows information we expect

Above can be performed with three `curl`'s and some `sed`-fu

SSO Test - Teardown

- Remove containers
- Dump container logs

LDAP Test

- Image must provide LDAP authentication
- Follows structure of SSO test (setup, execute, teardown)
- Containerized LDAP server with test data

LDAP Test - Snippet

```
do_curl_test_with_basic_auth \
 "fakeuser:fakepassword" \
 "401" \
 "fake user 'fakeuser' should not be able to authenticate"

do_curl_test_with_basic_auth \
 "testuser1:fakepassword" \
 "401" \
 "valid user 'testuser1' should not be able to authenticate with an invalid password"

do_curl_test_with_basic_auth \
 "testuser1:testpassword1" \
 "200" \
 "valid user 'testuser1' should be authorized to access the webapp due to nested e-groups"

do_curl_test_with_basic_auth \
 "testuser2:testpassword2" \
 "403" \
 "valid user 'testuser2' should not be authorized to access the webapp"
```

Outline - We Are Here

Scaling Pipeline to Multiple Versions of Image

- GitLab CI missing "build matrix" feature
- Extends/Include
  - Newer, not supported by GitLab Runner Exec
- Anchors/References
  - What we use

YAML Anchors/References - Concept

```yaml
.hidden: &anchor
  image: docker:18.09

my_job:
  # reference to anchor
  <<: *anchor
  script:
    - docker info
```

YAML Anchors/References - Usage

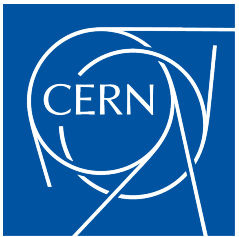
```
job_build_8_5_jdk7:
  variables:
    <<: *template_reference_common_variables_docker_dind
    <<: *template_reference_tomcat_8_5_jdk7_variables
  <<: *template_reference_build

job_build_9_0_jdk8:
  variables:
    <<: *template_reference_common_variables_docker_dind
    <<: *template_reference_tomcat_9_0_jdk8_variables
  <<: *template_reference_build
```

Outline - We Are Here

Future Improvements

- Unprivileged runner alternatives
- "ContainerStructureTest" tool
  - Recent Google tool similar to Goss
  - Basic tests runnable **without** Docker daemon (!)
- Docker image vulnerability scanning - "Clair"



www.cern.ch

Outline - We Are Here

References

- Repository for this presentation: `https://gitlab.cern.ch/tloekkeb/voxxeddays-cern-docker-image-testing-in-gitlab-ci`
- Goss: `https://github.com/aelsabbahy/goss`
- GitLab CI vulnerability scanning docs: `https://docs.gitlab.com/ee/user/application_security/container_scanning/`
- GitLab CI Docker-in-Docker docs: `https://docs.gitlab.com/ee/ci/docker/using_docker_build.html`
- ContainerStructureTests: `https://github.com/GoogleContainerTools/container-structure-test`
- GitLab Runner Exec: `https://docs.gitlab.com/runner/commands/#gitlab-runner-exec`
- Keycloak: `https://github.com/keycloak/keycloak`
- Travis CI build matrix: `https://docs.travis-ci.com/user/build-matrix/`

Features Used - Docker-in-Docker - Tricks

- docker:dind as a service - `alias: localhost` to make `docker run -p 8080:8080 nginx && curl localhost:8080` from the job container work as expected
- `docker build --cache-from` to manually specify image to cache layers from

YAML Anchors/References - Pitfall

```yaml
.hidden: &anchor
  image: docker:18.09
  variables:
    COMMON_VAR: ""

my_job:
  <<: *anchor
  script:
    - docker info
  # Will not work!
  variables:
    MY_JOB_VAR: ""
```

YAML Anchors/References - Pitfall - Solution

```yaml
.hidden_job: &anchor_job
  image: docker:18.09

.hidden_var: &anchor_var
  COMMON_VAR: ""

my_job:
  <<: *anchor_job
  script:
    - docker info
  variables:
    MY_JOB_VAR: ""
    <<: *anchor_var
```

YAML Template Example - Common all DinD jobs

```
.template_common_all_docker_dind_jobs: &template_reference_common_all_docker_dind_jobs
  image: docker:18.09
  tags:
    - db-runner-docker-privilege
  before_script:
    - docker info
    - docker login -u gitlab-ci-token -p $CI_BUILD_TOKEN $CI_REGISTRY
    - if [ -z $RUNNING_LOCALLY ]; then ln -s "$PWD" /artifacts && echo "CREATED SYMLINK"; fi
  services:
    - name: docker:18.09-dind
      alias: localhost

.template_common_variables_docker_dind: &template_reference_common_variables_docker_dind
  DOCKER_HOST: tcp://docker:2375/
  DOCKER_DRIVER: overlay2
```

YAML Template Example - Build

```
.template_build: &template_reference_build
  stage: build
  <<: *template_reference_common_all_docker_dind_jobs
  script:
    - docker pull ${PUSH_IMAGE_LATEST} || true
    -> docker build
      --cache-from ${PUSH_IMAGE_LATEST}
      --build-arg TOMCAT_MAJOR_VERSION="${TOMCAT_MAJOR_VERSION}"
      --build-arg JAVA_MAJOR_VERSION="${JAVA_MAJOR_VERSION}"
      -t ${TEST_IMAGE_TAG} .
    - docker save --output /artifacts/${TEST_IMAGE_TAR} ${TEST_IMAGE_TAG}
  artifacts:
    expire_in: '1 day'
    paths:
      - ${TEST_IMAGE_TAR}
```

YAML Template Example - Variables

```
.template_image_tag_variables: &template_reference_image_tag_variables
  TEST_IMAGE_TAG: test:${CI_COMMIT_REF}-${TOMCAT_MAJOR_VERSION}-${JAVA_MAJOR_VERSION}
  TEST_IMAGE_TAR: test:${CI_COMMIT_REF}-${TOMCAT_MAJOR_VERSION}-${JAVA_MAJOR_VERSION}.tar
  PUSH_IMAGE_LATEST: ${CI_REGISTRY_IMAGE}/${TOMCAT_FULL_VERSION}/${JAVA_FULL_VERSION}:latest

.template_tomcat_8_5_jdk7_variables: &template_reference_tomcat_8_5_jdk7_variables
  TOMCAT_MAJOR_VERSION: "8"
  JAVA_MAJOR_VERSION: "7"
  <<: *template_reference_image_tag_variables

.template_tomcat_9_0_jdk8_variables: &template_reference_tomcat_9_0_jdk8_variables
  TOMCAT_MAJOR_VERSION: "9"
  JAVA_MAJOR_VERSION: "8"
  <<: *template_reference_image_tag_variables
```

YAML Template Example - Usage

```
job_build_8_5_jdk7:
  variables:
    <<: *template_reference_common_variables_docker_dind
    <<: *template_reference_tomcat_8_5_jdk7_variables
  <<: *template_reference_build

job_build_9_0_jdk8:
  variables:
    <<: *template_reference_common_variables_docker_dind
    <<: *template_reference_tomcat_9_0_jdk8_variables
  <<: *template_reference_build
```

```
job_test_8_5_jdk7_goss:
  variables:
    <<: *template_reference_common_variables_docker_dind
    <<: *template_reference_tomcat_8_5_jdk7_variables
  <<: *template_reference_test_goss
  dependencies:
    - job_build_8_5_jdk7
```

```
job_test_9_0_jdk8_goss:
  variables:
    <<: *template_reference_common_variables_docker_dind
    <<: *template_reference_tomcat_9_0_jdk8_variables
  <<: *template_reference_test_goss
  dependencies:
    - job_build_9_0_jdk8
```

```
job_push_8_5_jdk7:
  variables:
    <<: *template_reference_common_variables_docker_dind
    <<: *template_reference_tomcat_8_5_jdk7_variables
  <<: *template_reference_push
  dependencies:
    - job_build_8_5_jdk7

job_push_9_0_jdk8:
  variables:
    <<: *template_reference_common_variables_docker_dind
    <<: *template_reference_tomcat_9_0_jdk8_variables
  <<: *template_reference_push
  dependencies:
    - job_build_9_0_jdk8
```