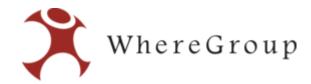
GitLab

Create your own version control with a CI depending on Docker



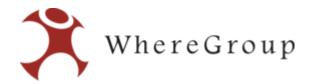
Thanks to Olaf and Peter to host this meetup!



- OpenSource
- WebGIS



- Products
- Consulting
- Development
- Support



Customers

- Deutsche Bahn
- Vattenfall
- Administration



Products

- Mapbender
- Metador
- Mops
- ...



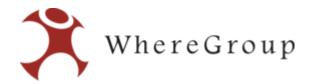
Involvement

- FOSSGIS
- FOSS4G
- OSGeo
- OGC



Locations

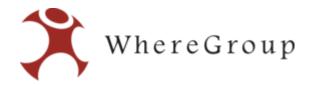
- Bonn
- Berlin
- Freiburg



By the way

We are hiring

www.wheregroup.com/de/jobs_karriere



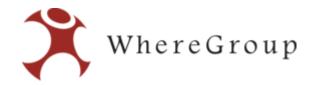
Speaker

• Arne Schubert



Topic

GitLab - Create your own version control with a CI depending on Docker



Pre-Requirements

- Docker
- docker-compose



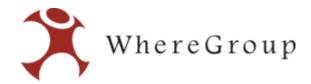
Pre-Requirements Ubuntu

```
apt-get update \
  && apt-get install -y \
    linux-image-extra-virtual \
    apt-transport-https \
    ca-certificates \
    curl \
    software-properties-common \
  && echo "Register docker software-repository..." \
  && curl -fsSL https://download.docker.com/linux/ubuntu/gpg | apt-ke
  && add-apt-repository \
    "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
    $(lsb release -cs) \
    stable" \
  && echo "Install docker..." \
  && apt-get update \
  && apt-get install -y docker-ce \
```



GitLab Community edition

GitLab is a web-based Git repository manager with wiki and issue tracking features, using an open source license



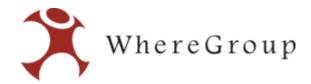
Prepare GitLab in docker-compose

```
version: '3'
services:
    gitlab:
        image: gitlab/gitlab-ce:latest
        volumes:
            - /var/run/docker.sock:/var/run/docker.sock
            - ./gitlab-ce:/etc/gitlab
            - gitlab-data:/var/opt/gitlab
            - gitlab-logs:/var/log/gitlab
        environment:
            GITLAB OMNIBUS CONFIG: "external url 'http://gitlab.acme
        ports:
            - 80:80
            - 443:443
            - 44:22
volumes:
```



SSH Port

Be careful with the SSH port 22!



Update permissions

If you run into access problems:

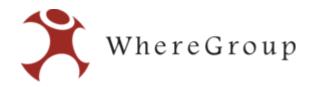
docker-compose exec gitlab update-permissions



Install GitLab without docker

about.gitlab.com/installation

```
apt-get install -y curl openssh-server ca-certificates
curl -sS https://packages.gitlab.com/install/repositories/gitlab/gitl
apt-get install gitlab-ce
gitlab-ctl reconfigure
```



Set a root user

Go to your GitLab server with a browser of choice and register the root user



Admin Runner

Runner with a connection to the docker ecosystem

Creates docker images for other CI runners



Add the runner in docker-compose

```
version: '3'
services:
    gitlab:
    # ...
    expose:
        - 80
        - 443
        - 22
    admin-runner:
        image: gitlab/gitlab-runner:latest
        volumes:
        - /var/run/docker.sock:/var/run/docker.sock
        - ./admin-runner/config:/etc/gitlab-runner
```



Register the runner

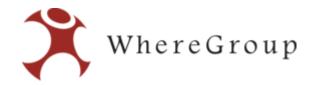
docker-compose exec admin-runner gitlab-runner register

Recommendation: Register this runner as a specific runner to a secured repository



Additional settings for the runner

```
[[runners]]
  [runners.docker]
   image = "acme-dind"
   allowed_images = ["acme*"]
   volumes = ["/cache", "/var/run/docker.sock:/var/run/docker.sock"
   pull_policy = "if-not-present"
  [runners.cache]
```



Dockerfile for "dind" in admin runner

```
FROM alpine:latest

MAINTAINER Arne Schubert <atd.schubert@gmail.com>

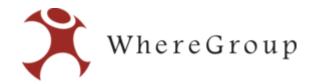
RUN set -x \
    && apk --no-cache add docker

VOLUME ["/var/run/docker.sock"]
```



Build it up!

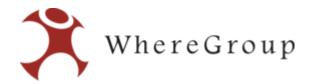
docker build -t acme-dind .



What means dind?

Docker in Docker

- Independent
- Dependent



Shared Runner

Runner to build projects from GitLab

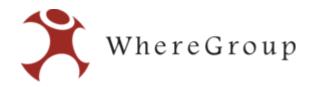


Add the runner in docker-compose



Register the runner

docker-compose exec shared-runner gitlab-runner register



Additional settings for the runner

```
[[runners]]
    [runners.docker]
    image = "acme/build"
    allowed_images = ["acme/*"]
    volumes = ["/cache" "/home/bob/.composer"]
    pull_policy = "if-not-present"
    [runners.cache]
```



Example for an docker image to build php projects

```
FROM php:latest
MAINTAINER Arne Schubert <atd.schubert@gmail.com>
RUN set -x \
  && apt-get update \
  && DEBIAN FRONTEND=noninteractive apt-get install --no-install-reco
    git \
    zip unzip ca-certificates \
    openssh-client \
    curl \
    libsqlite3-dev \
  && rm -rf /var/lib/apt/lists/* \
  && curl -o /usr/bin/composer https://getcomposer.org/composer.phar
  && chmod a+x /usr/bin/composer \
  && docker-php-source extract \
```



Build automated directly in GitLab

```
stages:
   - build

package:
   stage: build
   tags:
     - admin
   only:
     - master
   script:
     - docker build -t acme/build:latest .
```

Note: Use the admin-runner for that build



Create your dind image also from GitLab

You can create now the dind image the same way directly from GitLab!



Update GitLab

Just pull the images from docker-hub

docker-compose pull
docker-compose restart



Stages

- Package: in master (and develop)
- Test / Lint: on merge in develop and master
- Deploy: only for master



Runner

- shared: Default
- priority: Just for projects that needs a higher priority
- admin: Only for admin purpose
- {name}: Specific runners for specific projects (higher priority)



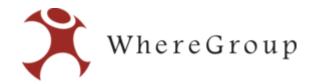
Artifacts

```
package:
    stage: build
    script:
        - echo "Do something..."
    artifacts:
        name: "${CI_JOB_NAME}_${CI_COMMIT_REF_NAME}"
        paths:
        - "${CI_PROJECT_DIR}/dist/*"
        expire_in: 1 week
```



GitLab-CE not on docker

- Run GitLab directly on your server (Port 22!)
- Run the runners in a docker swarm



Provisioning script

Use this gist for a automated installation

https://gist.github.com/atd-schubert/aba495fa500472d04113a50dc78a6f07



Last but not least

Checkout our GitHub - Accounts

@wheregroup

@yagajs



Last but not least

We hope you enjoy our meetup!

meetup@wheregroup.com



Let's discuss now

... while drinking some beer or limonade

meetup@wheregroup.com

