

GitLab

Create your own version control with a CI
depending on Docker



WhereGroup

WhereGroup

Thanks to Olaf and Peter to host this meetup!



WhereGroup

WhereGroup

- OpenSource
- WebGIS



WhereGroup

WhereGroup

- Products
- Consulting
- Development
- Support



WhereGroup

WhereGroup

Customers

- Deutsche Bahn
- Vattenfall
- Administration



WhereGroup

WhereGroup

Products

- Mapbender
- Metador
- Mops
- ...



WhereGroup

WhereGroup

Involvement

- FOSSGIS
- FOSS4G
- OSGeo
- OGC



WhereGroup

WhereGroup

Locations

- Bonn
- Berlin
- Freiburg



WhereGroup

WhereGroup

By the way

We are hiring

www.wherogroup.com/de/jobs_karriere



WhereGroup

Speaker

- Arne Schubert



WhereGroup

Topic

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



WhereGroup

Pre-Requirements

- Docker
- docker-compose



WhereGroup

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-key add - \  
  && 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 \  
  && systemctl enable docker
```



WhereGroup

GitLab Community edition

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



WhereGroup

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:
```



WhereGroup

SSH Port

Be careful with the SSH port 22!



WhereGroup

Update permissions

If you run into access problems:

```
docker-compose exec gitlab update-permissions
```



WhereGroup

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/gitlab-ce  
apt-get install gitlab-ce  
gitlab-ctl reconfigure
```



WhereGroup

Set a root user

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



WhereGroup

Admin Runner

Runner with a connection to the docker ecosystem

Creates docker images for other CI runners



WhereGroup

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
```



WhereGroup

Register the runner

```
docker-compose exec admin-runner gitlab-runner register
```

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



WhereGroup

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]
```



WhereGroup

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" ]
```



WhereGroup

Build it up!

```
docker build -t acme-dind .
```



WhereGroup

What means dind?

Docker in Docker

- Independent
- Dependent



WhereGroup

Shared Runner

Runner to build projects from GitLab



WhereGroup

Add the runner in docker-compose

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



WhereGroup

Register the runner

```
docker-compose exec shared-runner gitlab-runner register
```



WhereGroup

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]
```



WhereGroup

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-recommends \
    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 \
```



WhereGroup

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



WhereGroup

Create your dind image also from GitLab

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



WhereGroup

Update GitLab

Just pull the images from docker-hub

```
docker-compose pull  
docker-compose restart
```



WhereGroup

Stages

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



WhereGroup

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)



WhereGroup

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
```



WhereGroup

GitLab-CE not on docker

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



WhereGroup

Provisioning script

Use this gist for a automated installation

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



WhereGroup

Last but not least

Checkout our GitHub - Accounts

@wherogroup

@yagajs



WhereGroup

Last but not least

We hope you enjoy our meetup!

meetup@wherengroup.com



WhereGroup

Let's discuss now

... while drinking some beer or limonade !

meetup@wherogroup.com



WhereGroup