

CI/CD mit Openshift

Thomas Herzog B.Sc / Phillip Wurm B.Sc

January 18, 2018

Abstract

Dieses Dokument beinhaltet die Dokumentation der *CI/CD*-Umgebung, die in Openshift aufgesetzt wurde.

1 Entwickler Setup

Dieser Abschnitt beschreibt das Aufsetzen einer lokalen Entwicklungsumgebung für die Entwicklung mit Openshift. Es wird davon ausgegangen, dass auf einem Linux System gearbeitet wird.

1.1 Docker Setup

Dieser Abschnitt behandelt, dass Einrichten von Docker für die Verwendung von Openshift. Es muss eine aktuelle Version von Docker installiert sein.

```
// 1. Define insecure registry, which is used by openshift
//     depending on your linux distribution
INSECURE_REGISTRY='--insecure-registry 172.30.0.0/16'

// 2. Reload the docker service
sudo systemctl daemon-reload

// 3.Restart the docker service
sudo systemctl restart docker
```

Die ungesicherte Registry wird von Openshift dazu verwendet, um in Openshift gebaute Docker Images zu verwalten.

1.2 Firewall Setup

Dieser Abschnitt behandelt das Einrichten der Firewall für die Verwendung von Openshift.

```
// 1. Check docker bridge subnet
docker network inspect
    -f "{{range .IPAM.Config }}{{ .Subnet }}{{end}}" bridge

// 2. Create new firewall zone
firewall-cmd --permanent --new-zone dockerc

// 3. Add docker bridge network as source
firewall-cmd --permanent --zone dockerc --add-source 172.17.0.0/16

// 4. Add all ports docker and openshift needs
firewall-cmd --permanent --zone dockerc --add-port 8443/tcp
firewall-cmd --permanent --zone dockerc --add-port 53/udp
firewall-cmd --permanent --zone dockerc --add-port 8053/udp

// 5. Reload the firewall rules
firewall-cmd --reload
```

Ohne diese Firewall Einstellungen kann kein Docker Container, der in Openshift läuft, DNS Namen auflösen, oder auf das Internet zugreifen.

1.3 Openshift Setup

Dieser Abschnitt beschreibt das Einrichten des lokalen Openshift *Clusters*.

Es werden folgende Ressourcen benötigt, die aus dem Internet heruntergeladen werden können.

1. ***Openshift Client Tools***

<https://developers.redhat.com/products/openshift/download/>

Es wird die Version *3.5.5.31.24* benötigt, die unter *älter Versionen* gefunden werden kann.

Es wird ein aktiver JBoss *Developer Account* vorausgesetzt.

2. ***openshift-client-wrapper***

<https://github.com/openshift-evangelists/oc-cluster-wrapper/releases/tag/0.9.3>:

Bei *oc-cluster-wrapper* handelt es sich um ein Shell-Skript, welches das Arbeiten mit oc erleichtert.

Das *oc* Binary sowie das Skript *oc-cluster-wrapper* müssen in den *PATH* mitaufgenommen werden. Das Skript *oc-cluster-wrapper* verwendet das *oc* Binary, das mit *oc* über den *PATH* angesprochen werden kann.

1.3.1 Kommandos

Dieser Abschnitt beschreibt die Kommandos, die für das Kontrollieren des *Cluster*.

```
# Create or start persistent profile for local cluster named 'ci'
oc-cluster-wrapper up ci
```

```
# Stop the local cluster of profile 'ci'
oc-cluster-wrapper down ci
```

```
# Delete profile ci and all related cluster data
oc-cluster-wrapper destroy ci
```

Wenn beim Starten folgende Fehlermeldung auftritt, dann liegt es daran, dass die Konfiguration nicht mehr gültig, da Sie sich z.B. in einem anderen Netz befinden.

```
oc cluster up --version v3.5.5.31
--image registry.access.redhat.com/openshift3/ose
--public-hostname 127.0.0.1
--routing-suffix apps.127.0.0.1.nip.io
--host-data-dir /home/het/.oc/profiles/ci/data
--host-config-dir /home/het/.oc/profiles/ci/config
--host-pv-dir /home/het/.oc/profiles/ci/pv -
-use-existing-config -e TZ=CET
-- Checking OpenShift client ... OK

...

Finding server IP ...
Using 10.29.18.80 as the server IP
-- Starting OpenShift container ... FAIL
Error: Docker run error rc=2
Details:
Image: registry.access.redhat.com/openshift3/ose:v3.5.5.31
Entrypoint: [/bin/bash]
Command: [-c for name in 10.29.18.80 het.linux.gepardec.com; kma
do ls /var/lib/origin/openshift.local.config/node-$name &> /dev/null
&& echo $name && break; done]
```

Kopieren Sie das *Shell*-Kommando ***oc cluster up ...*** und entfernen Sie das Argument ***-use-existing-config***. Ohne das Argument ***-use-existing-config*** wird eine neue Konfiguration erstellt.

2 *Build Server*

Dieser Abschnitt behandelt die Infrastruktur des *Build-Server* Projekts in Openshift. Die Ressourcen für den *Build Server* werden im *Repository buildserver*¹ verwaltet.

2.1 *Templates*

Dieser Abschnitt behandelt die Openshift *Templates*, welche die Services für die *Build-Server* Infrastruktur definiert. Die Openshift *Templates* beinhalten alle Definitionen wie z.B. *BuildConfigurations* und *Deployments*, die Aspekte des *Core Concepts*² von Kubernetes und Openshift sind.

Diese Auflistung beschreibt die implementierten *Templates*:

1. Im *Template jenkins-slaves.yml* werden die für Jenkins zur Verfügung gestellten *Slave-Container* verwaltet.
2. Im *Template jenkins.yml* wird der Jenkins Service verwaltet.
3. Im *Template nexus.yml* wird der Nexus3 Service verwaltet.
4. Im *Template pipeline.yml* wird für das Anlegen einer Openshift *Pipeline* verwendet.

2.2 *Skripte*

Dieser Abschnitt behandelt die Skripten, die für das Verwalten des *Clusters* verwendet werden. Mit der Applikation *oc* kann mit dem *Cluster* interagiert werden, wie z.B. PProjekte erstellen/löschen, oder Applikation in Projekten anlegen/löschen. Damit der *Build Server* einfach erstellt oder gelöscht werden kann, sind für die Services und für den *Build Server* Skripte erstellt worden, die alle nötigen Kommandos beinhalten.

Auf dem Level der Skripten wird eine Datei namens *.openshift-env* erwartet, die Umgebungsvariablen definiert, die von Skripten verwendet wird.

Diese Auflistung beschreibt die implementierten *Skripte*:

1. Im *Skript openshift-jenkins.sh* sind alle Jenkins Service und Jenkins *Slave* spezifischen Funktionen implementiert.
2. Im *Skript openshift-nexus.sh* sind alle Jenkins Service und Jenkins *Slave* spezifischen Funktionen implementiert.

¹<https://github.com/OpenshiftCICD/buildserver>

²https://docs.openshift.com/container-platform/3.5/architecture/core_concepts/index.html

3. Im Skript ***openshift-buildserver.sh*** sind alle Funktionalitäten für das Verwalten des *Build Servers* implementiert.
4. Im Skript ***openshift-secrets.sh*** sind alle Funktionalitäten für das Verwalten von *Secrets* implementiert. Siehe Abschnitt 2.5 für eine genauere Beschreibung der verwendeten *Secrets*.

2.3 Aufbau

Dieser Abschnitt behandelt den Aufbau des *Build Servers*.

2.4 Upgrade/Downgrade

Dieser Abschnitt behandelt die *Update*-Szenarien für den *Build Server*. Es gibt drei Szenarien für das Updaten des *Build Servers*.

1. Bei ***Änderung der Quelltexte*** wie *S2I Builds*³ oder *Dockerfiles*, muss der Docker *Container* neu gebaut und der Service aktualisiert werden.
2. Bei ***Änderung der Templates***, muss der Service aktualisiert werden.
3. Bei ***Änderung der Docker Images***, die Services beinhalten oder Basisimages darstellen, muss der Service aktualisiert werden.

Openshift erlaubt es bei *DeploymentConfigs* und *BuildConfigs* *Trigger*⁴ zu definieren, die bei Ereignissen wie *Github Commit*, *ImageChange* oder *ConfigChange* ausgelöst werden. Die folgenden Abschnitte beschreiben die verwendeten *Trigger*.

2.5 Secrets

Dieser Abschnitt behandelt die verwendeten *Secrets* des *Build Servers*.

³<https://github.com/openshift/source-to-image>

⁴https://docs.openshift.com/container-platform/3.5/dev_guide/builds/triggering_builds.html