Die eigene Docker Registry

»Vertraue, aber prüfe nach«

Russisches Sprichwort

Lernziele

★ Sie haben eine eigene Registry aufgesetzt.

Zeitlicher Ablauf

- ★ Die Registry im Detail
- ★ Setup – Insecure Registry
- ★ Übung: Insecure Registry aufsetzen
- ★ Lokaler Registry-Mirror
- ★ Übung: Insecure Registry mit UI aufsetzen
- ★ Reflexion
- ★ Lernzielkontrolle

Die Registry im Detail (1)

- ★ Die **Registry** stellt prinzipiell und stark vereinfacht nichts anderes dar als ein **Speicher- und Content-Delivery-System**, das selbst als Container-Instanz auf dem Host ausgeführt wird.
- ★ Die **Registry** speichert und verwaltet die in ihr gehosteten Docker Images in einer **hierarchischen Struktur**.
- ★ Der **Docker-Anwender** bzw. die CLI (oder sonstige externe Clients) kommunizieren über die üblichen **docker [image] pull**- und **push**-Befehle mit der API der entsprechenden Registry
- ★ Der **Default Storage Driver** für die Registry ist das **lokale Posix-Dateisystem** (*filesystem*), das für Entwicklungsumgebungen oder kleinere Implementierungen geeignet ist.
- ★ In Verbindung mit **SAN**-Anbindungen oder Posix-kompatiblem, skalierbarem (Shared) Software Defined Storage, können auch größere Deployments **problemlos bedient** werden.
- ★ Zusätzliche Cloud-basierte Storage Driver für die Registry wie **S3** (»3 x S«: Simple Storage Services), **Microsoft Azure, OpenStack Swift, GCS (Google Cloud Services)** und andere werden ebenfalls **unterstützt**, daneben können über eine offene Storage-API eigene Treiberimplementierungen eingeklinkt werden.

Quelle: Buch Skalierbare Container-Infrastrukturen, Kapitel 6.1

Die Registry im Detail (2)

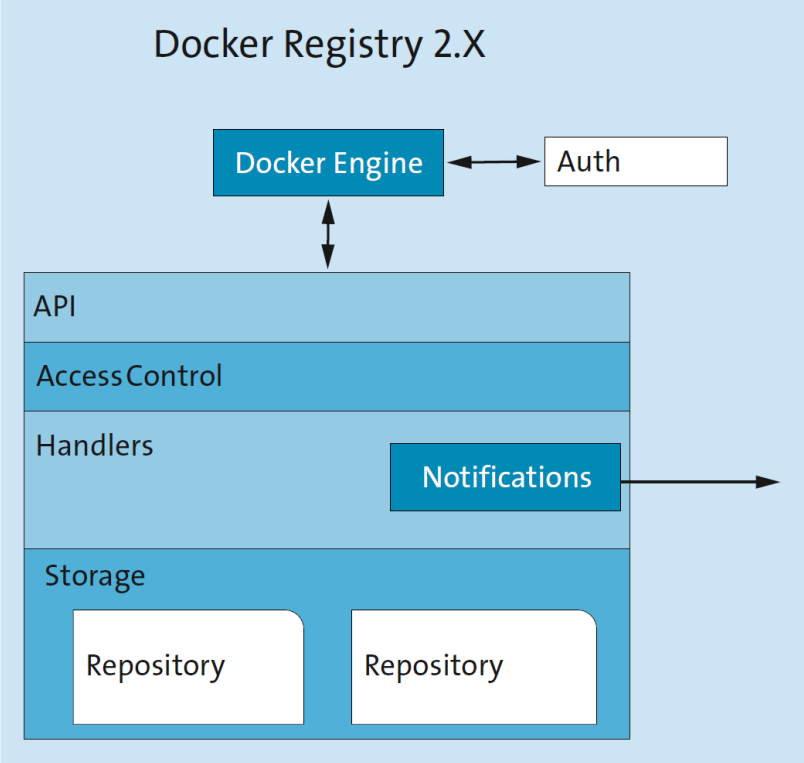

Abbildung 6.1 Stark vereinfachter funktionaler Aufbau einer Docker Registry

- ★ Neben der reinen Storage-Implementierung zur Ablage der Images muss der Registry aber auch noch eine Instanz vorgeschaltet sein, die sich protokollbasiert (HTTP[S]) um die Kommunikation mit der Außenwelt kümmert, einen Index der vorhandenen Images enthält und zusätzlich authentifikationsrelevante Belange abwickelt.
- ★ Dies übernimmt in der Regel ein Webserver, der im optimalen Fall ausschließlich via TLS kommuniziert und eine Authentifizierung via LDAP zusammen mit einem RBAC(Role Based Access Control)-Modell implementiert.

Quelle: Buch Skalierbare Container-Infrastrukturen, Kapitel 6.1

Setup – Insecure Registry

- ★ Den Docker Daemon so konfigurieren, dass wir auf die (Remote-)Registry via http (insecure) zugreifen können.
  - Variante per daemon.json-Konfiguration:
    - ★ `{ "insecure-registries": ["localhost:5000"] }`
  - Alternativ können wir diesen Schalter auch direkt über den dockerd-Parameter `--insecure-registry` setzen.
  - Bei «Docker for Windows» oder «Mac» ist diese Einstellung via Assistent möglich.
- ★ Anschliessend kann die Registry gestartet werden:
  - `docker run -d -p 5000:5000 --restart always --name registry registry:2.6`
- ★ Offizielle Docker Registry: https://hub.docker.com/_/registry/

Quelle: Buch Skalierbare Container-Infrastrukturen, Kapitel 6.2

Übung: Insecure Registry aufsetzen (05-1-Registry)

Übung: Insecure Registry

Startet eine lokale Registry und legt ein Image darin ab.

Zuerst Ausgabe der Hilfeinformationen zur Registry, damit wir die Funktionen kennen

In [ ]:
```
! docker container run --rm registry:2.6 --help
```

Starten der Insecure Registry

In [ ]:
```
! docker run -d -p 5000:5000 --restart always --name registry registry:2.6
```

Ausgabe der Registry Konfiguration

In [ ]:
```
! docker exec registry cat /etc/docker/registry/config.yml
```

Image in lokaler Registry ablegen und Resultat mittels `docker image ls` überprüfen

In [ ]:
```
! docker pull ubuntu
! docker tag ubuntu localhost:5000/ubuntu
! docker push localhost:5000/ubuntu
! docker image ls
```

Beide Images weglöschen, Resultat überprüfen und Image wieder aus lokaler Registry holen

In [ ]:
```
! docker rmi ubuntu localhost:5000/ubuntu
! docker image ls
! docker pull localhost:5000/ubuntu
! docker image ls
```

Lokaler Registry-Mirror

- ★ Ein Registry-Mirror kann eine deutliche Entlastung auf mehreren Ebenen einbringen:
  - Sofern der Zugriff aller Docker Clients auf nur eine zentrale Registry erfolgt, muss diese logischerweise alle anfragenden Clients bedienen.
  - Eine Möglichkeit, sowohl Ausfallsicherheit als auch etwas Lastverteilung einzuziehen, kann ein Registry-Mirror sein.
- ★ Dabei wird, vereinfacht ausgedrückt, eine Registry als »Proxy-Cache« vor die eigentliche Registry geschaltet. Die in ihr abgelegten Images können nun von anderen Docker Clients genutzt werden.
- ★ Bei einem Pull eines Docker Clients wird als Erstes der Registry-Mirror abgefragt. Kann er kein passendes Image anbieten, geht der Request an die reguläre Registry.
- ★ Dazu braucht es eine Registry und die entsprechende Registry-Mirror-Direktive (für dockerd z. B.: **--registry-mirror <liste>**).
- ★ Bei «Docker for Windows» oder «Mac» ist diese Einstellung via Assistent möglich.

Voraussetzungen Trusted Registry

- Eindeutiger DNS Name (darf auch intern sein).
- Server Zertifikat von CA abgeleitet, Let's Encrypt Zertifikate sind zulässig.
- Abschaltung der insecure-registry.
- Client Zertifikat von CA abgeleitet.

★ **Registry-Authentifizierung**

- Simple/Basic Authentication via htpasswd
- Token-Authentifizierung (fehleranfällig)
- Zentrale Registry-Authentifizierung via LDAP/TLS

Hinweis

Als CA-Zertifikate werden nur Dateien mit der Endung *.crt* berücksichtigt. Zertifikats-/Key-Dateien können als *.cert*- und *.key*- oder als *.pem*-Dateien vorliegen.

Quelle: Buch Skalierbare Container-Infrastrukturen, Kapitel 6.5 – 6.7

Übung: Insecure Registry mit UI aufsetzen

- ★ `kubeps.bat` oder `kubesh.bat` im Verzeichnis lernkube starten.
- ★ Docker Registry und Frontend starten (via Kubernetes)
  - `kubectl apply -f duk/registry/registry2.yaml`
  - `kubectl apply -f duk/registry/registry2-frontend.yaml`
- ★ Image in lokaler Registry ablegen
  - `docker pull ubuntu`
  - `docker tag ubuntu localhost:32500/ubuntu`
  - `docker push localhost:32500/ubuntu`
- ★ Docker Registry Frontend aufrufen
  - http://localhost:32580

Weitere Registries

- ★ Nexus3 - von den Maven Leuten
- ★ Harbor - aus KubeWeekly
- ★ JFrog Artifactory
- ★ What is the best alternative to Docker Registry

Reflexion

- In Produktiven Umgebungen sollte die Verwendung einer »trusted«-Registry, für die Ablage der Images, eine zwingende Voraussetzung sein.
- Docker stellt dazu leider nur sehr rudimentäre Hilfsmittel (ihre Registry) zur Verfügung, weshalb man ohne Zusatzaufwand nicht auskommt.

- **Und die Registry stellt in jedem Unternehmen eine der zentralen Komponenten dar, über die alle Docker Hosts mit Images betankt werden. Fällt sie aus, laufen alle angeschlossenen Docker Hosts beim nächsten Pull oder Build ins Leere.**

Lernzielkontrolle

★ Sie haben eine eigene Registry aufgesetzt.