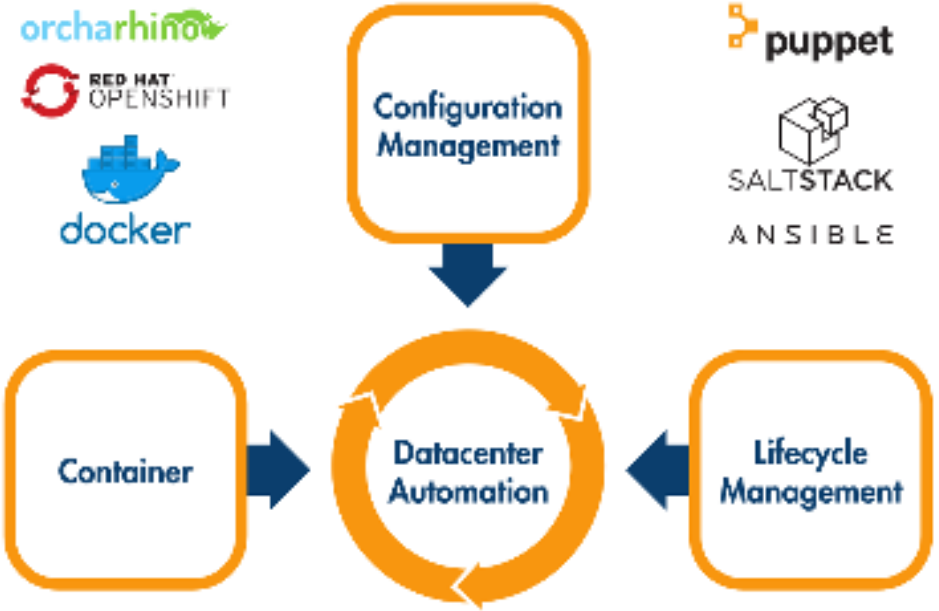
Community-Plattform

mit Docker und Kubernetes

Dr. Jan Bundesmann

16.03.2019

Wer bin ich und was ist Atix?



CONSULTING ENGINEERING SUPPORT

orcharhino

DEPLOY
RUN
CONTROL

#kubernetes #container #community

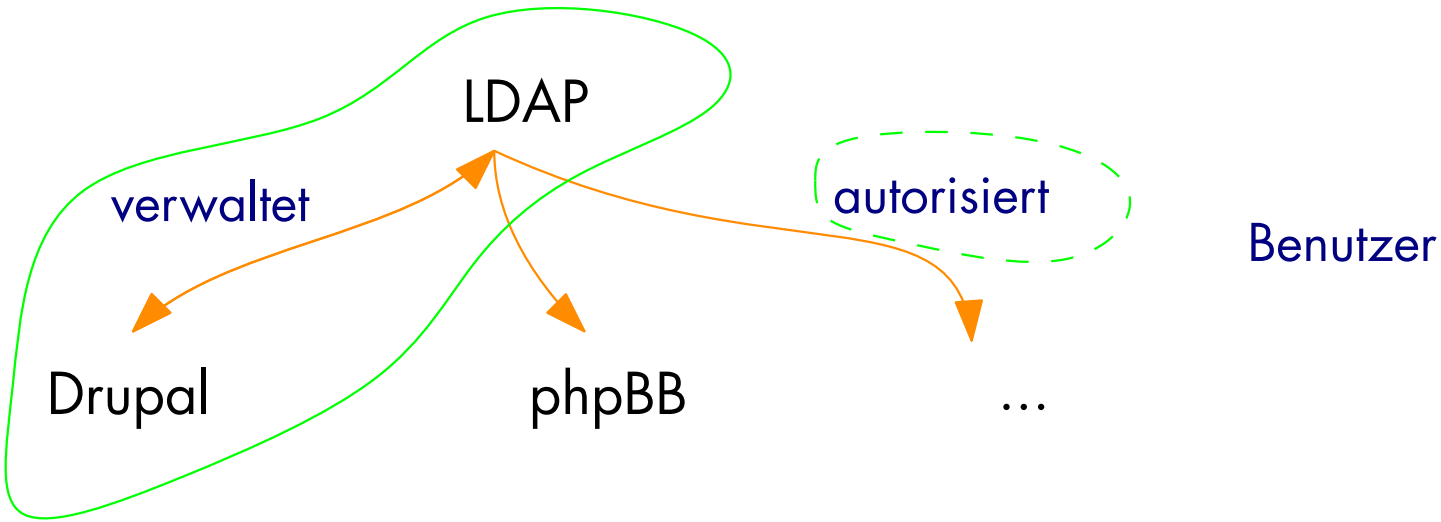
Inhaltsverzeichnis

Ziel

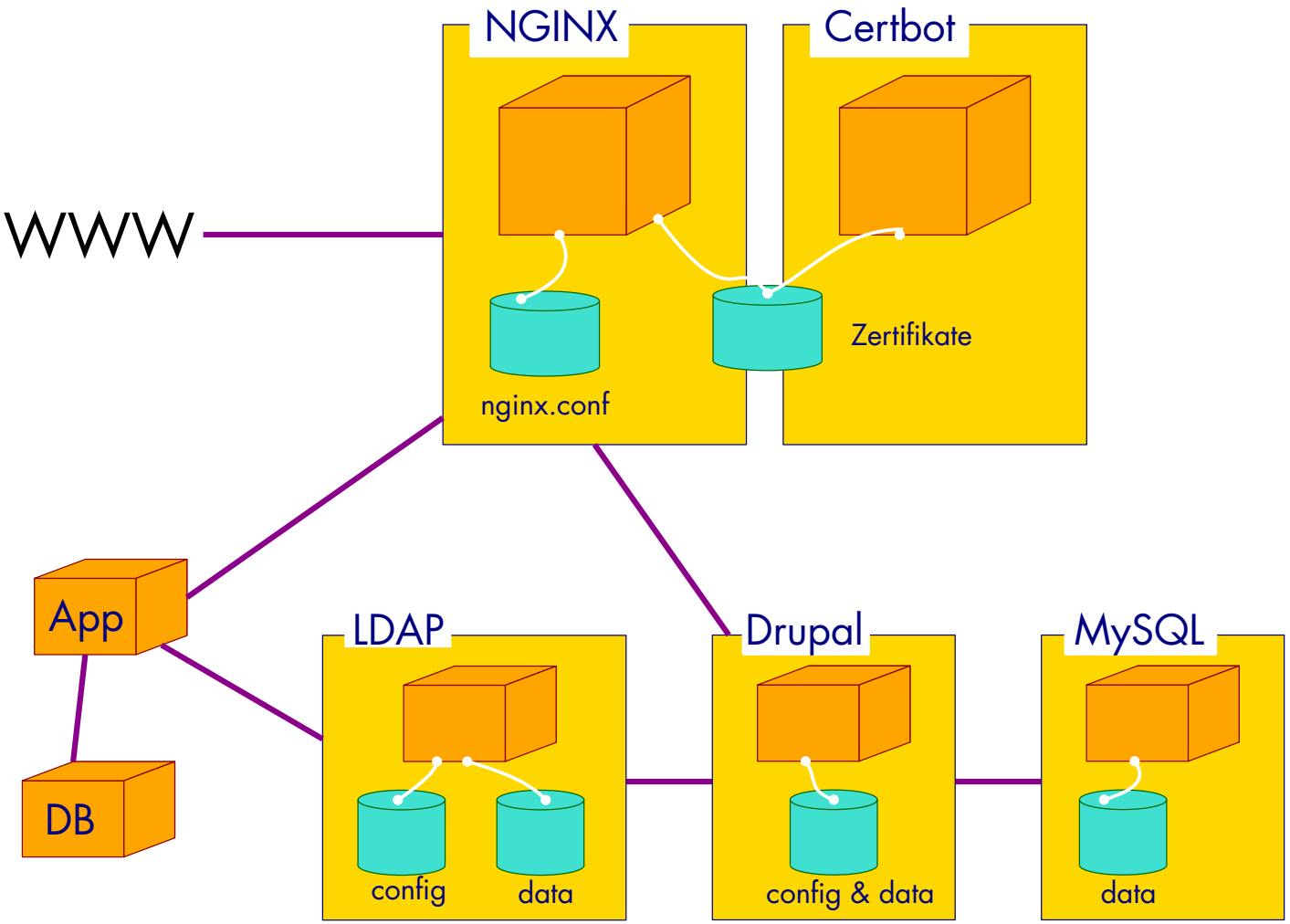
- Website
- angemeldete Nutzer
- Kommunikationsmöglichkeit für Nutzer
- zusammengesetzt aus Standards

Problem: keine Verbindung im reinen LAMP-Stack

Vorstellung System & Eingrenzen auf Kern



Kern als Container



Kubernetes statt Docker

- Randbedingung: kein eigener Server ⇒ “as a Service”
- häufiger Kubernetes als Docker as a Service

Was ist Kubernetes?

- Resourcen-Management-System:
  - Compute
  - Storage
  - Netzwerk
  - Cluster
- Containerverwaltungssystem:
  - Storage und Netzwerk stark abstrahiert
  - Compute immer in Form von Containern
  - Containerlogik inhärent

vgl. Torque/Maui, HTCondor, Nomad, Mesos

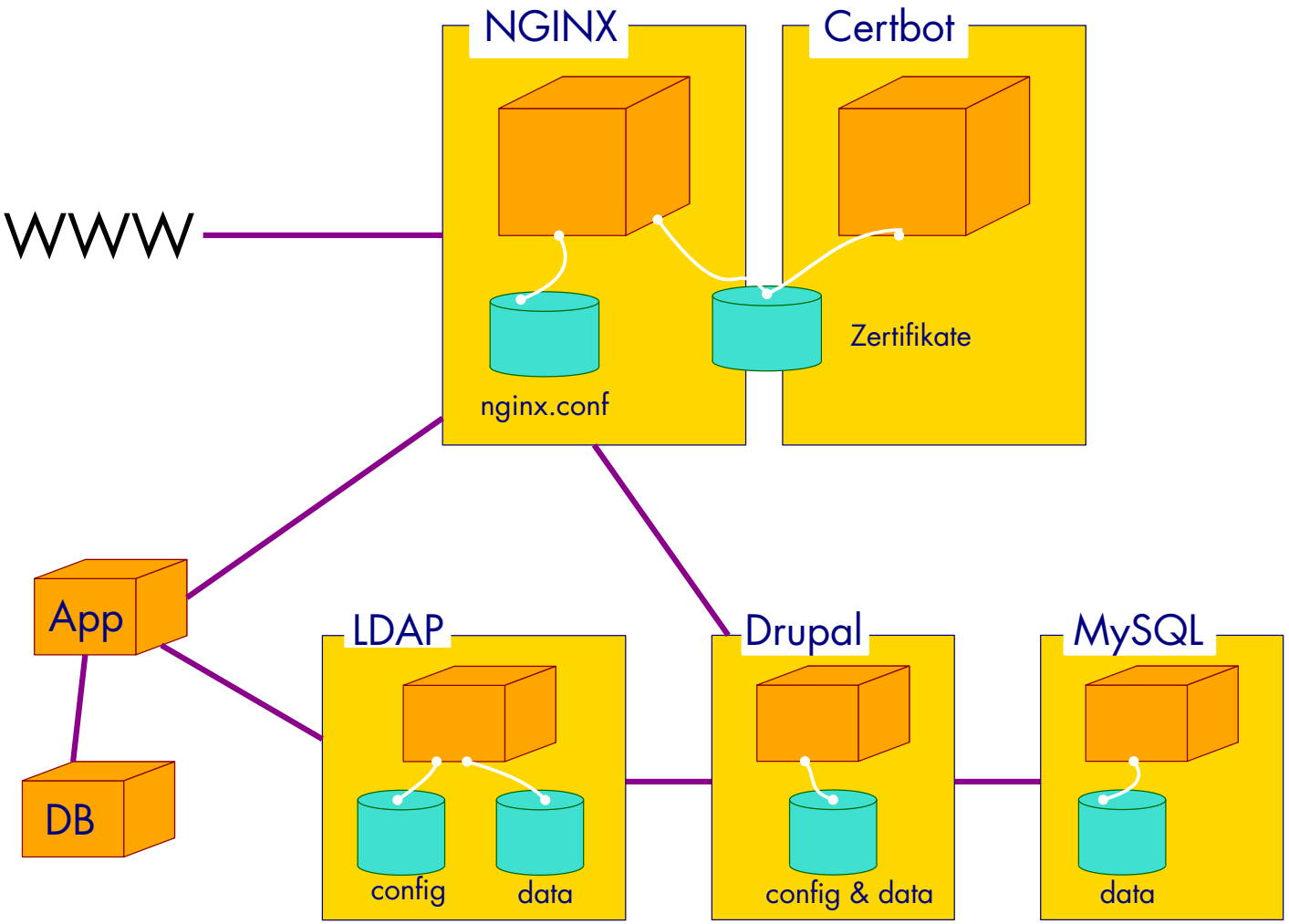
typische Resourcen in K8s I

- `Pod`
  - Containeraggregat auf einem Node
  - Bsp.: App mit Monitoring & Istio Sidecar
- `Deployments`
  - langlebige Pods
  - replizieren und verteilen Pods
- `Jobs` starten Pods x Mal

typische Resourcen in K8s II

- `Services`
  - binden sich an `Deployments`
  - machen TCP-Socket (`hostname:port`) im Cluster oder nach außen verfügbar
- `PersistentVolumeClaims` fragen nach Storage
- `ConfigMaps` für Konfigurationsdateien und Umgebungsvariablen

Core-System als Ausblick



Let's Encrypt Zertifikat mit certbot

```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx
spec:
  ports:
    - port: 80
      name: http
    - port: 443
      name: https
  selector:
    app: nginx
  type: LoadBalancer
```

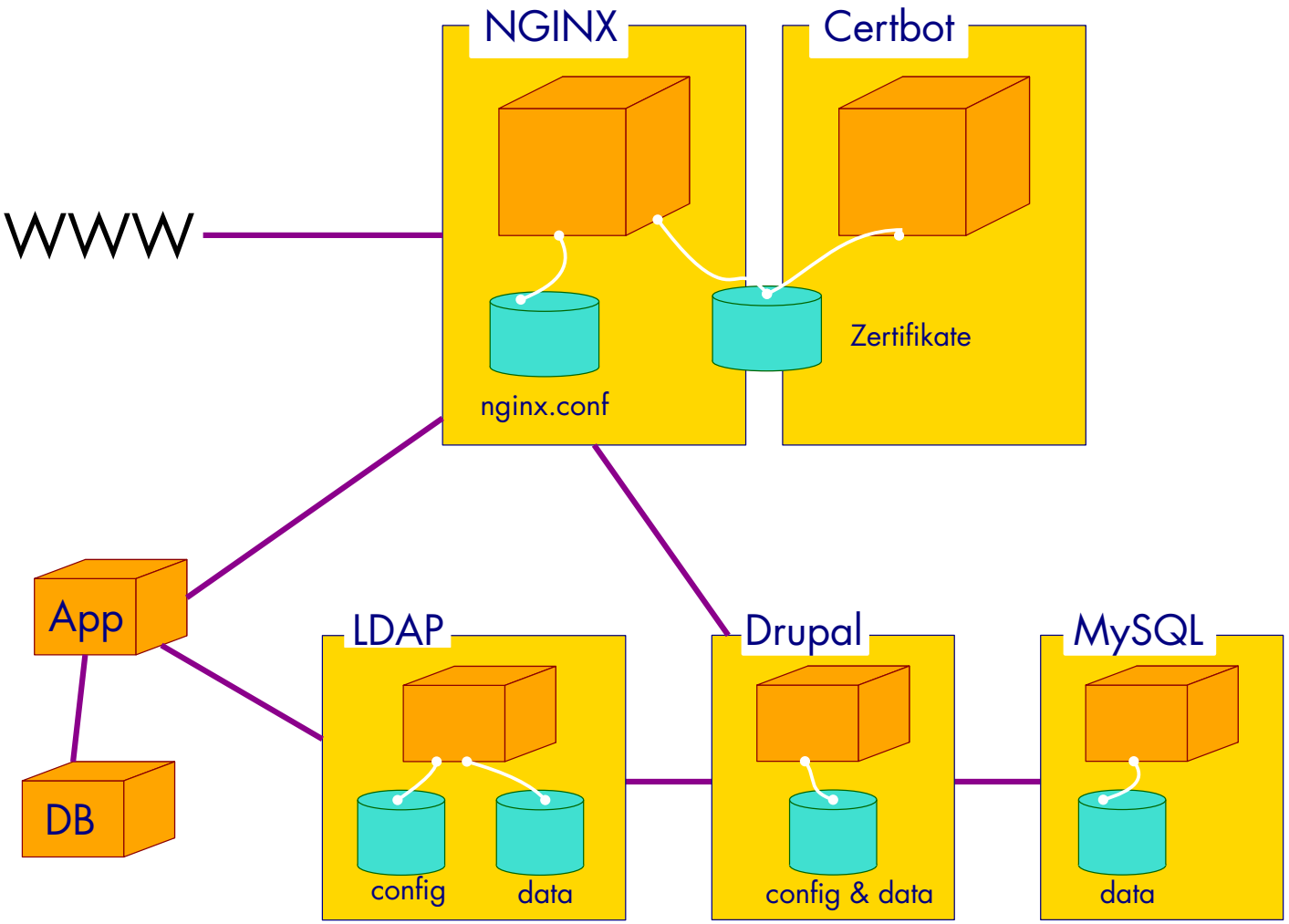
```
kubectl create -f my_service.yaml
```

PersistentVolumeClaim

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: claim-nginx
spec:
    storageClassName: do-block-storage
    accessModes:
        - ReadWriteOnce
    resources:
        requests:
            storage: 1Gi
```

```
kubectl create -f my_pvc.yaml
```

Erinnerung: Core-System



Let's Encrypt Zertifikat mit certbot

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: claim-nginx
spec:
    storageClassName: do-block-
          storage
    accessModes:
        - ReadWriteOnce
    resources:
        requests:
            storage: 1Gi
---
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  replicas: 1
  template:
```

```
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  volumes:
    - name: nginx-conf
      configMap:
        name: nginx-config
    - name: nginx-data
      persistentVolumeClaim:
        claimName: claim-nginx
  containers:
    - name: nginx
      image: nginx
      ports:
        - containerPort: 80
          name: "http"
        - containerPort: 443
          name: "https"
      volumeMounts:
        - name: nginx-conf
          mountPath: "/etc/nginx/conf.d/default.conf"
          subPath: "nginx.conf"
        - name: nginx-data
          mountPath: '/usr/share/nginx/html'
          subPath: 'www-data'
        - name: nginx-data
          mountPath: '/etc/letsencrypt'
          subPath: 'letsencrypt'
```

ohne Zertifikat

nginx.conf:

```
server {
    listen 80;
    server_name clt.demo.bundesmann.it;

    location / {
      root   /usr/share/nginx/html;
      index  index.html index.htm;
    }
}
```

```
kubectl create configmap nginx-config \
  --from-file nginx.conf
kubectl create -f deployment_nginx.yaml
```

Let's Encrypt Zertifikat mit certbot

```
apiVersion: batch/v1
kind: Job
metadata:
    name: certbot
spec:
  template:
    spec:
      volumes:
        - name: nginx-data
          persistentVolumeClaim:
            claimName: claim-nginx
      containers:
        - name: certbot
          image: certbot/certbot
          command: ["certbot"]
          args: ["certonly", "--webroot", "-w", "/usr/share/nginx/html",
          "-d", "clt.demo.bundesmann.it", "--email",
          "bundesmann@atix.de", "-n", "--agree-tos", "-v",
          #"bundesmann@atix.de", "-n", "--agree-tos", "-v", "--dry-run",
          ]
          volumeMounts:
            - name: nginx-data
              mountPath: '/usr/share/nginx/html'
              subPath: 'www-data'
            - name: nginx-data
              mountPath: '/etc/letsencrypt'
              subPath: 'letsencrypt'
      restartPolicy: Never
```

mit Zertifikat

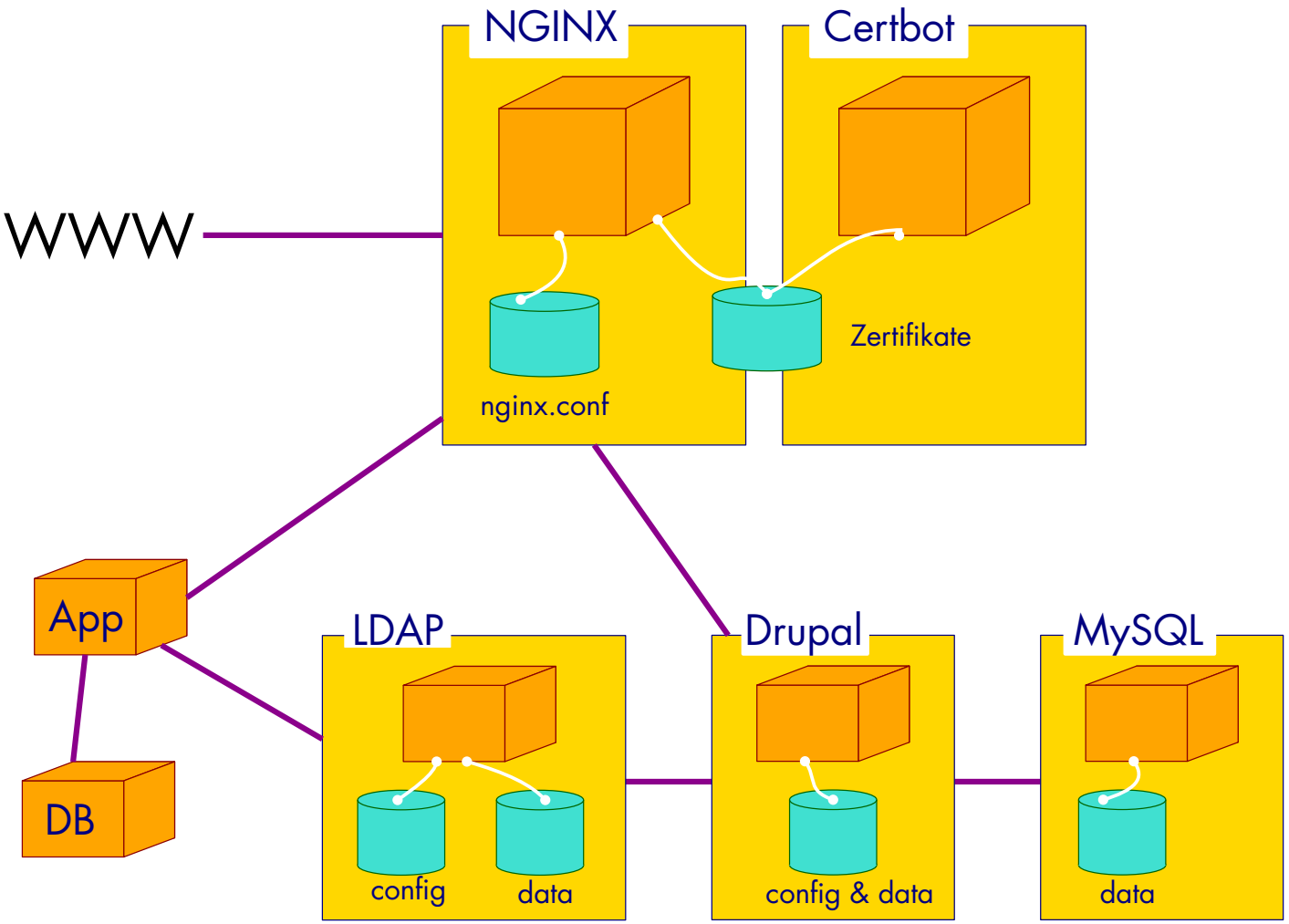
```
server {
    listen 80;
    listen 443 ssl;
    server_name clt.demo.bundesmann.it;

    ssl_certificate /etc/letsencrypt/archive/clt.demo.bundesmann.it/cert1.pem;
    ssl_certificate_key /etc/letsencrypt/archive/clt.demo.bundesmann.it/privkey1.pem;

    location / {
      root   /usr/share/nginx/html;
      index  index.html index.htm;
    }
}
```

```
kubectl delete configmap nginx-config
kubectl create configmap nginx-config \
  --from-file nginx.conf
kubectl delete pod nginx-...
```

Und nochmal: Wie weit sind wir jetzt?



LDAP

eigener Container oder vorhandene Instanz?

- Service **`clt-ldap-server`**
- PersistentVolumeClaim **`clt-volume-ldap`**
- ConfigMap mit Prepopulate-Daten **`clt-variables-ldap`**
- Deployment

LDAP – prepopulate

01_basis.ldif

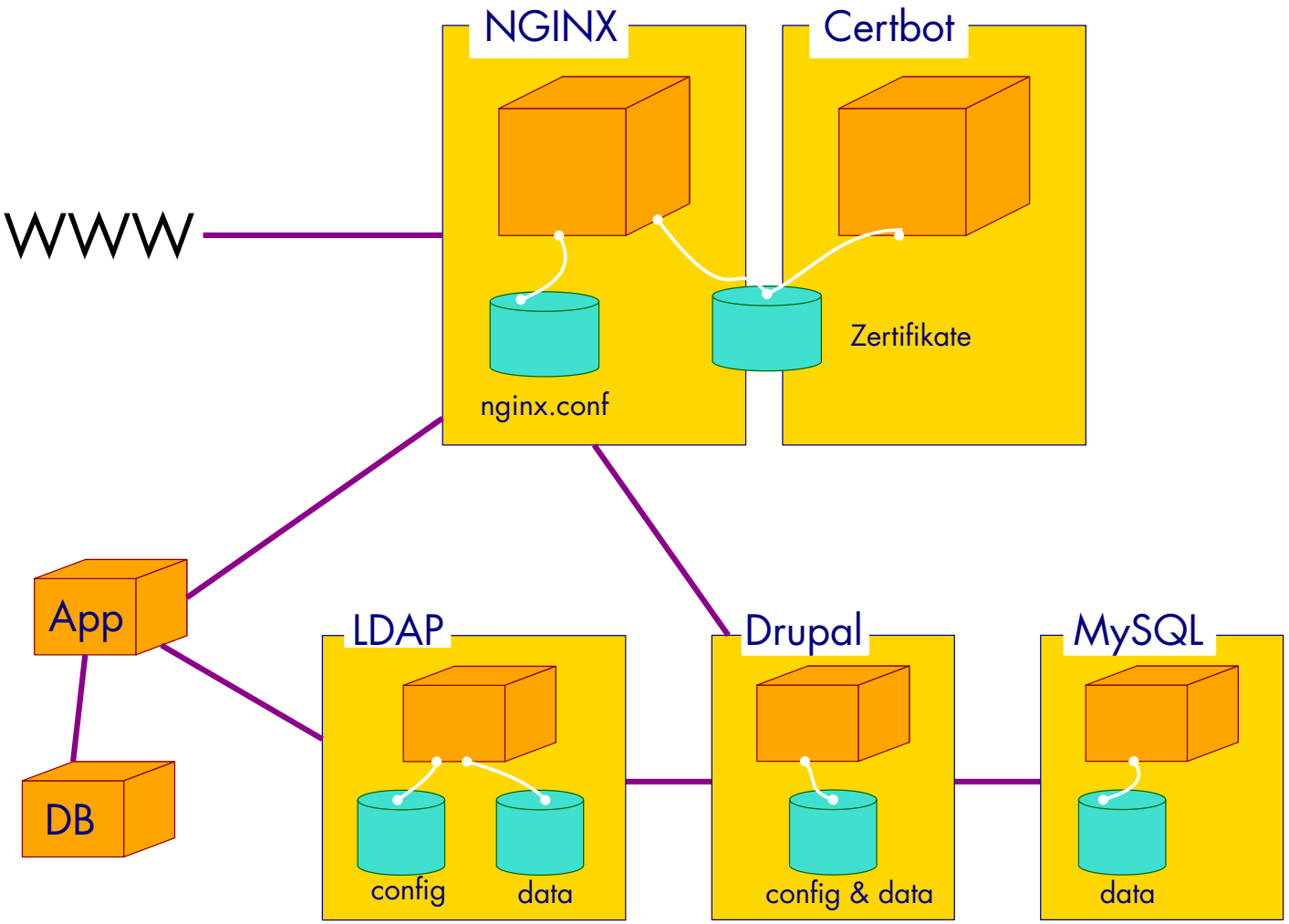
```
dn: ou=people,dc=clt
objectclass: top
objectclass: organizationalUnit
ou: people
```

02_users.ldif

```
dn: cn=admin,ou=people,dc=clt
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: admin
sn: admin
mail: bundesmann@atix.de
```

```
kubectl create configmap clt-variables-ldap --from-file 01_basis.
ldif --from-file 02_users.ldif
```

Und jetzt?


NGINX
Certbot
WWW
Zertifikate
nginx.conf
App
LDAP
Drupal
MySQL
DB
config
data
config & data
data

Drupal-Kern: Image ausstatten

LDAP & Mail:

```
FROM drupal:7-apache

# Some software we need for the later extensions of php
RUN apt-get update && apt-get purge -y sendmail \
  && apt-get install -y libldap2-dev libmcrypt-dev ssmtp vim \
  && rm -rf /var/lib/apt/lists/*

# We want to authorize using a central LDAP directory
RUN docker-php-ext-configure ldap --with-libdir=lib/x86_64-linux-gnu/ \
    && docker-php-ext-install ldap

# The query-user needs a password stored in the drupal db, mcrypt helps
    to encrypt it
RUN docker-php-ext-install mcrypt

# German language
WORKDIR /var/www/html/profiles/standard/translations
RUN curl -s -o drupal-7.64.de.po \
  https://ftp.drupal.org/files/translations/7.x/drupal/drupal-7.64.de.po

# build modules
WORKDIR /var/www/html/modules
```

https://github.com/knoppi/drupal_docker

https://hub.docker.com/r/knoppi/drupal

Drupal-Kern: Module I

```
# smtp module
ENV url https://ftp.drupal.org/files/projects/smtp-7.x-1.7.tar.gz
ENV md5_sum c0184179267654a739306af63fbf267f
RUN curl -s -o module.tar.gz $url \
    && echo $md5_sum module.tar.gz | md5sum -c \
    && tar zxf module.tar.gz && rm module.tar.gz

# devel module
ENV url https://ftp.drupal.org/files/projects/devel-7.x-1.6.tar.gz
ENV md5_sum 1176b4c249ef0c398a763c6ffcc9b18c
RUN curl -s -o module.tar.gz $url \
    && echo $md5_sum module.tar.gz | md5sum -c \
    && tar zxf module.tar.gz && rm module.tar.gz

# entity module
ENV url https://ftp.drupal.org/files/projects/entity-7.x-1.9.tar.gz
ENV md5_sum 793870ebcaa31da748e165d470c0b9bb
RUN curl -s -o module.tar.gz $url \
    && echo $md5_sum module.tar.gz | md5sum -c \
    && tar zxf module.tar.gz && rm module.tar.gz

# LDAP module
ENV url https://ftp.drupal.org/files/projects/ldap-7.x-2.3.tar.gz
ENV md5_sum cb7b235060185caf8da03fd5be5b7917
RUN curl -s -o module.tar.gz $url \
    && echo $md5_sum module.tar.gz | md5sum -c \
    && tar zxf module.tar.gz && rm module.tar.gz
```

Drupal-Kern: Module II

```
# ctools module
ENV url https://ftp.drupal.org/files/projects/ctools-7.x-1.14.tar.gz
ENV md5_sum 88dbe403ecfe2fe434f4237e5fd5ec67
RUN curl -s -o module.tar.gz $url \
    && echo $md5_sum module.tar.gz | md5sum -c \
    && tar zxf module.tar.gz && rm module.tar.gz

# drush module
WORKDIR /opt
ENV url https://ftp.drupal.org/files/projects/drush-7.x-5.9.tar.gz
ENV md5_sum 70feb5cb95e7995c58cbf709a6d01312
RUN curl -s -o module.tar.gz $url \
    && echo $md5_sum module.tar.gz | md5sum -c \
    && tar zxf module.tar.gz && rm module.tar.gz

RUN chmod u+x drush/drush \
    && ln -s /opt/drush/drush /usr/bin/drush

RUN chown -R www-data:www-data /opt /var/www/html

# create a backup of the image contents of specific folders
WORKDIR /bak
RUN mv /var/www/html/sites /var/www/html/themes ./

WORKDIR /var/www/html/
COPY entrypoint.sh /entrypoint.sh
ENTRYPOINT ["/entrypoint.sh"]
```

Drupal starten und verfügbar machen

- Services: **`clt-drupal-app`** & **`clt-drupal-db`**
- PersistentVolumeClaim: **`clt-drupal`**
- Deployments: **`drupal-app`** & **`mysql`**
- Anpassung in nginx.conf

```
location / {
  proxy_pass http://clt-drupal-app/;

  proxy_set_header Host $http_host;
  proxy_set_header X-Real-IP $remote_addr;
  proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
  proxy_set_header X-Forwarded-Proto $scheme;
}
```

`https://cs.demo.bundesmann.it`

LDAP aktivieren & einrichten

Configuration → LDAP Configuration

NING FROM DRUPAL TO LDAP MAPPINGS:

Source Drupal User Attribute			Target LDAP Token
(Select "user tokens" to use token field)	Source Drupal User tokens such as "[propertyname]". "[field.field_fname] [field.field_lname]". Constants such as "from_drupal" or "18" should not be enclosed in []	Convert From Binary	Use singular token format such as [sn], [givenName], etc.
Property: Email			[mail]
Property: Username			[cn]
-- user tokens --	cn=[property.name],ou=people,o=orklager		[dn]
-- user tokens --	top		[objectclass:0]
-- user tokens --	person		[objectclass:1]
-- user tokens --	organizationalPerson		[objectclass:2]
-- user tokens --	inetOrgPerson		[objectclass:3]
-- user tokens --	Drupal Provisioned LDAP Account		[description]
Property: Username			[uid]
Property: Username			[sn]
Pwd: User or Random			[userPassword]

Anflaschen: phpBB

- Subdirectory hinter Reverse Proxy kompliziert
- eigenes Image mit Subdirectory `forum` hartkodiert
- LDAP Anbindung sehr einfach
- noch zu tun: lokale Registrierung verbieten

`https://cs.demo.bundesmann.it/forum`

kein SSO!

Anflanschen: RocketChat

- Problem wieder: Reverse Proxy & Subdirectory
- Lösung `ROOT_URL` und `proxy_pass` im NGINX auf gleichen Pfad zeigen lassen
- LDAP Anbindung unkompliziert

`https://cs.demo.bundesmann.it/chat`

Anflanschen: Nextcloud

Zusammenfassung

- Glückliche Orks?
- Kernschwierigkeiten: LDAP-Integration & Proxy
- Kubernetes as a Service

Und nun?

- Einfach ausprobieren per Minikube oder Schnupperangebote
- Drupal 8
- mehr Apps anhängen
- mehr automatisieren
  - Helm-Chart
  - Installation der Anwendungen
- einfacher Backupmechanismus

Das Beste zum Schluss

©Moritz Jendral

Twitter:
@janbundesmann
@superknoppi

Mail: bundesmann@atix.de

… oder halt am Atix-Stand

github.com/knoppi/drupal_docker
hub.docker.com/r/knoppi/drupal