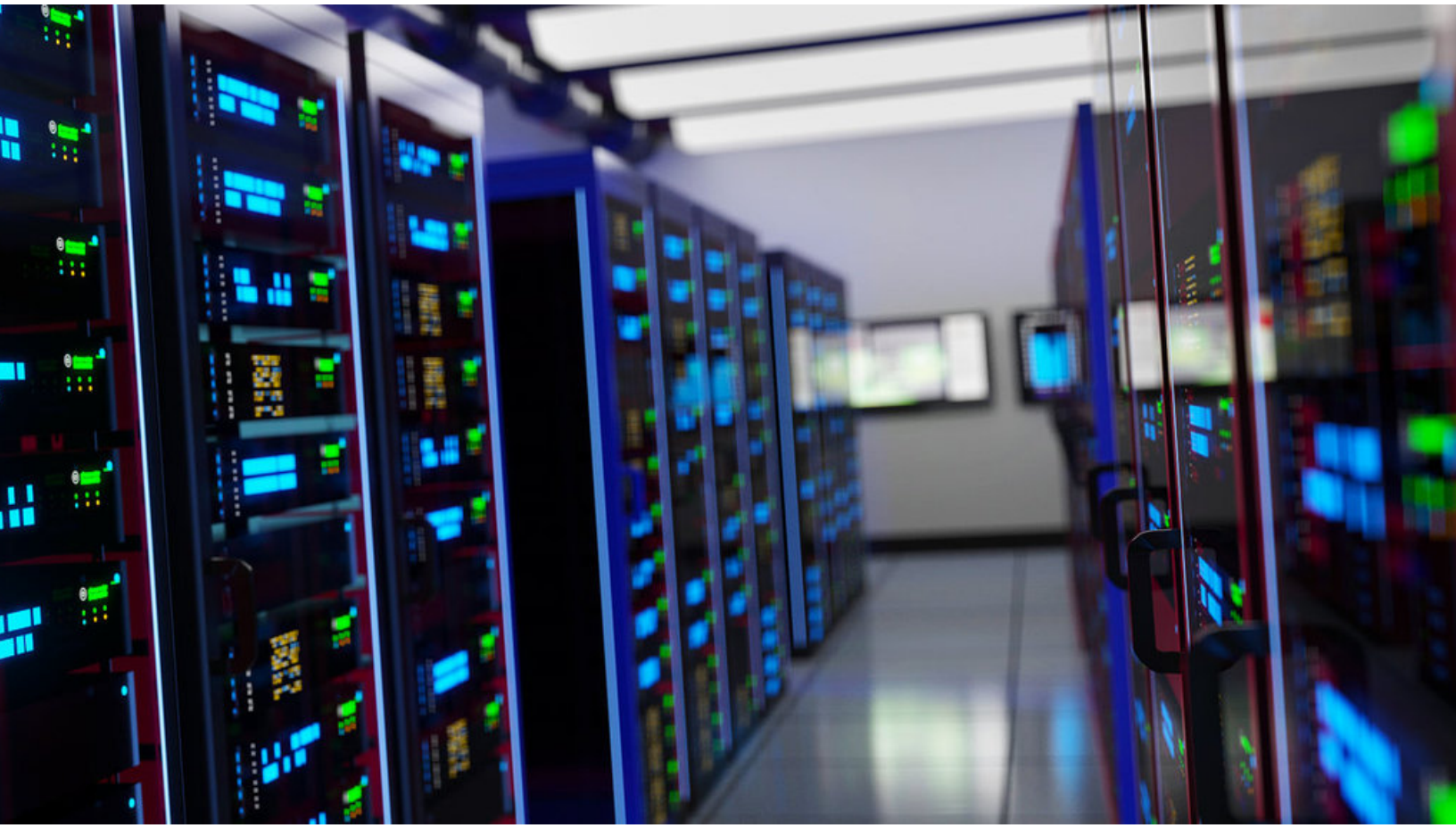
2019



LB01 - M239

M239 - INTERNETSERVER IN BETRIEB NEHMEN

CHRISTOPH KÜNDIG, JOHANNES FLASCHBERGER

TBZ – TECHNISCHE BERUFSSCHULE ZÜRICH

Versionshistorie

Version	Datum	Autor	Änderungen
1.0	18.02.19	Kündig	Initial Version Dokument
1.1	04.03.19	Flaschberger	Ist-Situation erfasst und Pflichtenheft erstellt
1.2	04.03.19	Kündig	Checkliste erstellt
1.3	11.03.19	Flaschberger	Planung erstellt und Lastprofil
1.4	11.03.19	Kündig	Ziele für Mailserver ergänzen
1.5	18.03.19	Flaschberger & Kündig	Vorgehen dokumentiert von Realisationsphase
1.6	25.03.19	Kündig	Testfälle erstellt.
2.0	01.04.19	Flaschberger & Kündig	Erklärung und Funktionsweise Protokolle erstellt.

Inhaltsverzeichnis

1 Anforderungen

1.1 Ist-Situation

Beispielszenario:
Die Firma «Restart IT» bietet IT Support an und sind auf KMU's und Privatpersonen spezialisiert. Ihre Kunden können über eine E-Mail-Adresse oder per Telefon die Firma erreichen. Aktuell sind 15 Mitarbeiter in der Firma tätig. Momentan wird noch kein File Server eingesetzt und die Mitarbeiter tauschen die Dateien entweder per Mail oder USB-Stick aus. Es hat lediglich ein Mitarbeiter das nötige Knowhow um die Webseite zu bearbeiten, da diese noch manuell mit HTML, CSS und JS entwickelt wurde.

Der Webserver und der Mailserver werden intern auf einem Server betrieben. Da dieser Server mittlerweile in die Jahre gekommen ist und nicht mehr stabil läuft möchte der Chef, dass sie auf einem neuen Server die gesamte Webseite und Mailserver neu aufsetzten. Zusätzlich um die Arbeit der Mitarbeiter zu erleichtern wurde beschlossen, dass die Anleitungen und Dokumente der Mitarbeiter lokal auf einem Fileserver gespeichert werden, damit alle darauf zugreifen können. Für die Service-Techniker die Unterwegs sind, sollten diese Dateien auch verfügbar sein. Der Chef hat zudem erwähnt, dass es ihm wichtig ist, dass die Firma einfach aktuell gehalten werden kann.

1.2 Pflichtenheft

Die Firma möchte den Telefonsupport abbauen, da dieser sehr zeitaufwändig ist. Der Chef möchte den Onlinechat Support einführen. Zudem sollten die Anleitungen und Dokumente von allem Mitarbeiter erreichbar sein. Die Webseite soll von den Mitarbeitern verwaltet werden können, damit die Kunden auch dort schon Hilfestellungen bei Problemen finden können.

1.2.1 Anforderungen

- Webserver
    - Einfaches bearbeiten der Webseite
    - Online Chat Funktion
    - Hilfeartikel sind online verfügbar
    - Soll skalierbar sein
    - Sicherheit sollte nicht vernachlässigt werden
    - 100 User können gleichzeitig auf die Webseite zugreifen

- Mailserver
    - Mails können versendet und empfangen werden
    - Virenverseuchte Anhänge und weitere Gefahren sollen gefiltert werden

- Fileserver
    - Anleitungen und Dokumente können in einen gemeinsamen Share abgelegt werden
    - Jeder Mitarbeiter hat zusätzlich noch 100 GB Speicher für seine eigenen Dateien
    - Es kann auch von extern auf die Dateien zugegriffen werden
    - Sicherheit sollte nicht vernachlässigt werden

1.2.2 Sicherheitsaspekte

Folgende Sicherheitsaspekte sollen beachtet werden:

- Datenbanken sind von aussen nicht erreichbar
- Fileserver und Webseite sollen mit Zweifaktor Authentifizierung geschützt sein.
- Nur die Ports 80, 443, 993 und 465 sollen von aussen erreichbar sein.

1.2.3 Checkliste

- Webseite kann über HTTPS erreicht werden.
- Mails können versendet werden.
- 100 User können gleichzeitig auf die Webseite zugreifen.
- User können über den Onlinechat Kontakt mit den Mitarbeitern aufnehmen.
- Die Webseite kann leicht über ein CMS verwaltet werden.
- Der Zugriff auf die MySQL Datenbank ist nicht möglich von aussen.
- Alle Mitarbeiter können auf die Cloud zugreifen.

1.3 Lastprofil

Es wurde auf der alten Website mithilfe von Google Analytics ein Lastprofil erstellt. Das Ergebnis ist auf dem Diagramm ersichtlich.

Unique Visitor = Ein Unique Visitor, zu Deutsch „einzelner Besucher“, ist die Gesamtzahl der Besucher auf einer Website innerhalb eines bestimmten Zeitraumes. Dabei wird jeder einzelne Besucher der Webseite innerhalb dieses Zeitraums auch nur einmal gezählt - unabhängig davon, wie oft er die Seite aufruft.

1.4 Verfügbarkeit

Anhand des Lastprofils können wir ableiten, dass während dem Zeitraum 06 Uhr – 22 Uhr die meisten User auf die Webseite zugreifen. Wir möchten eine Verfügbarkeit von 99.8 % haben. Das bedeutet, dass in einer Woche die Webseite maximal 20 Minuten down sein kann. Für Wartungsarbeiten werden wir die Zeit ausserhalb des angegebenen Zeitraumes verwenden.

Pro Tag	Pro Woche	Pro Monat	Pro Jahr
2,88 Minuten	20,16 Minuten	86,4 Minuten	1.051,2 Minuten

2 Planung

2.1 Zeitplan

Wann?	Was?	Wer?
18.02.19	Dokumentation erstellen	Kündig
04.03.19	Ist-Situation erfasst und Pflichtenheft erstellt	Flaschberger
04.03.19	Checkliste erstellt	Kündig
11.03.19	Zeitplanung erstellt	Flaschberger und Kündig
18.03.19	Evaluation der Variante(n) für Linux	Flaschberger
18.03.19	Evaluation der Variante(n) für Windows	Kündig
25.03.19	Realisationsphase ➔ Umsetzen	Flaschberger und Kündig
25.03.19	Testfälle erfassen	Kündig
01.04.19	Realisationsphase ➔ Dokumentieren ➔ Resultate festhalten Kontrolle durchführen	Flaschberger und Kündig
15.04.19	Auswertung	Flaschberger
15.04.19	Präsentation erstellen und Vorbereitung Live-Demo	Flaschberger und Kündig

3 Entscheiden

3.1 Variante 1 Linux

Auf Linux gibt es viele Möglichkeiten die Anforderungen umzusetzen.
Wir werden uns mit Docker auseinandersetzen. Was ist eigentlich Docker? Nun es gibt virtuelle Maschinen und Container. Bei den virtuellen Maschinen sorgt ein Hypervisor dafür, dass die physischen Ressourcen unterhalb des Betriebssystems aufgeteilt werden. Bei den Containern wird auf Betriebssystemebene virtualisiert. Die Container laufen dann sozusagen wie als Prozess. Sie benötigen auch keine zusätzlichen Gastbetriebssysteme.

Vorteile von Docker
+ Schnellere Skalierbarkeit
+ Schnellere Erstellung der Container
+ Geringerer CPU Overhead
+ Besser im Nachbauen von Umgebungen (Produktion, Test und Entwicklungsumgebung)

Nachteile von Docker
- Schlechtere Isolierung im Gegensatz zu virtuelle Maschinen
- Daten innerhalb des Containers gehen verloren, wenn diese nicht auf dem Host gespeichert werden
- Nur CLI Applikation sind dafür gedacht innerhalb eines Containers ausgeführt zu werden.

Eingesetzte Container Images mit Begründung:

- Wordpress
    - Einfach zum Änderungen vornehmen
    - Grosse Community
    - Viele Themes
    - Viele Plugins

- Seafile
    - Übersichtliche Dateiverwaltung
    - Web-App, Mobile-App und Desktop-App inklusive
    - Performanter im Vergleich zu anderen Cloud Lösungen wie Owncloud oder Nextcloud

- MySQL
    - Wird benötigt für Wordpress und Seafile

- Mailcow
    - Setzt Postfix und Dovecot ein
    - Microsoft Active Sync wird unterstütz
    - Autodiscovery wird unterstütz

- Traefik
    - Einfacher Reverse Proxy um alles hinter einer öffentlichen IP-Adresse laufen zu lassen.

3.2 Variante 2 Windows

Es gibt auch für Windows verschiedene Webserver. Die bekannten Beispiele hierfür sind Apache, XAMPP und Microsoft IIS. Da wir hier bei der zweiten Variante ein komplett anderer Ansatz als bei der ersten Variante angehen, wird der Webauftritt unserer Firma unter Windows mit IIS bereitgestellt.
Als Mailserver wird hMailServer verwendet.

Vorteile von Microsoft IIS
+ Einfachere Integration mit anderen Microsoft Produkten
+ Einfache Verwaltung von mehreren Domains

Nachteile von Microsoft IIS
- Sehr teure Anschaffungskosten (Windows Server Lizenz)
- Grosse Sicherheitslücken
- Graphische Oberfläche (Teils komplizierter als über CLI, wie bei Linux)

3.3 Evaluation der Umsetzungsvariante

Da bei Linux die Installation mit Docker viel einfacher von Statten geht als bei Windows, haben wir uns für die Linux-Variante entschieden. Was noch dazu kommt, dass die Windows-Variante, je nach Art des Betriebes viel zu teuer wäre.

4 Realisieren

4.1 Vorgehen (Schritt für Schritt)

4.1.1 Variante 1 Linux

Um mit Docker zu arbeiten, muss Docker zuerst einmal installiert werden. Diese Step by Step Anleitung wurde auf Ubuntu 18.04 durchgeführt.

Installation von Docker
Mit diesem Befehl wird mit Curl ein Shell Skript von get.docker.com heruntergeladen und direkt zu sh gepipt.

```
curl -fsSL https://get.docker.com | sh
```

Als nächstes laden wir noch Docker-Compose herunter, da dies der Umgang mit Docker Container um vieles vereinfacht.

```
sudo curl -L "https://github.com/docker/compose/releases/download/1.24.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

Nach dem Download müssen die Berechtigungen noch angepasst werden.

```
sudo chmod +x /usr/local/bin/docker-compose
```

4.1.1.1 Webserver (Wordpress + MySQL)

Als nächstes erstellen wir eine Docker-Compose Datei in der wir alles Benötigte definieren. Die Datei muss als .yaml Datei abgespeichert werden, da in Docker-Compose alles mit YAML gemacht wird.

Beispiel von https://docs.docker.com/compose/wordpress/

```
version: '3.3'

services:
  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: somewordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress

  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    ports:
      - "8000:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
      WORDPRESS_DB_NAME: wordpress
volumes:
   db_data: {}
```

Beschreibung des Inhaltes:
Als erstes wird die **Version** des Compose files angegeben. Danach kommen die **Service** Definition. Hier werden die verwendeten Container definiert. Auf der nächsten Zeile wird einen Namen für den Container festgelegt, in diesem Fall **db**. Dann wird festgelegt welches **Image** verwendet werden soll, in diesem Fall mysql:5.7. Nach dem Doppelpunkt den Versions Tag festgelegt werden des Images. Auf hub.docker.com kann man die Tags einsehen. Mit **Volumes** kann ein lokales Directory in den Container eingebunden werden, dies hat den Vorteil, dass die Daten von dem Container ausserhalb des Containers gespeichert werden und somit der Container gelöscht werden kann ohne Datenverlust. **Restart**: always bedeutet, dass die Container auch gestartet werden, wenn Docker gestartet wird. Danach folgen Image spezifische **Umgebungsvariablen**, hier sind das die Passwörter für MySQL. Der nächste Abschnitt definiert den Container für Worpress. Hier werden noch zusätzlich Ports nach aussen freigegeben unter dem Abschnitt **ports.** Links ist der Externe Port welcher von aussen erreichbar sein wird und rechts der Interne Port sprich der Port der Applikation.

Mit diesem Befehl kann nun die Erstellung der Container gestartet werden. Sollte ein anderen Name als docker-compose.yaml verwendet werden muss dieser mit dem Parameter -f <Dateiname>vor dem Parameter up angegeben werden.

```
docker-compose up
```

Es ist auch möglich diesen Befehl im Detached Modus zu starten. Hierzu verwenden sie den -d Parameter nach dem Parameter up.

```
docker-compose up -d
```

Will man die Container wieder stoppen und gleich automatisch entfernen kann dieser Befehl verwendet werden.

```
docker-compose down
```

Um bereits gestoppte Container zu entfernen, kann mit dem Parameter rm gearbeitet werden.

```
docker-compose rm
```

Vorgehensweise WordPress nach Installation:

1. Admin Account festlegen.
2. Themes installieren.
3. Blog Funktionen einschränken.
4. Startseite und Supportseiten erstellen.
5. Onlinechat Plugin installieren und mit Account verbinden.
6. Benutzerauthentifizierung anpassen.

4.1.1.2 Mailserver (Mailcow)

Als erstes navigieren wir in den add-on Applikation Order von Linux /opt.

```
cd /opt
```

Dann klonen wir das Github Repository von Mailcow. Darin befinden sich alle benötigten Dateien und Konfigurationen.

```
git clone https://github.com/mailcow/mailcow-dockerized
```

Nun begeben wir uns in das neu geklonte Repository.

```
cd mailcow-dockerized
```

Hier starten wir ein Skript für das Erstellen der Konfiguration.
Es werden beim Ausführen des Skripts folgende Fragen gestellt:
Hostname (FQDN): mail.flaschberger.ch
Timezone [Etc/UTC]: Europe/Zurich
Installed memory is <= 2.5 GiB. It is recommended to disable ClamAV to prevent out-of-memory situations.
ClamAV can be re-enabled by setting SKIP_CLAMD=n in mailcow.conf.
Do you want to disable ClamAV now? [Y/n] y

```
./generate_config.sh
```

Das Skript hat nun die Datei mailcow.conf erstellt. In dieser Datei deaktivieren wir noch, dass Lets Encrypt verwendet werden soll für das erhalten eines SSL Zertifikats. Hierfür suchen wir nach SKIP_LETS_ENCRYPT= und setzten den Wert auf y. In dieser Datei können auch die verwendeten Ports geändert werden.

```
Vi mailcow.conf
```

Und dann laden wir die benötigten Images herunter. Dies ist nicht notwendig aber beschleunigt das spätere erstellen der Container.

```
docker-compose pull
```

Wenn alles heruntergeladen wurde, dann sollte das Output so aussehen:

```
Pulling unbound-mailcow   ... done
Pulling mysql-mailcow     ... done
Pulling redis-mailcow     ... done
Pulling clamd-mailcow     ... done
Pulling php-fpm-mailcow   ... done
Pulling sogo-mailcow      ... done
Pulling dovecot-mailcow   ... done
Pulling postfix-mailcow   ... done
Pulling memcached-mailcow ... done
Pulling nginx-mailcow     ... done
Pulling rspamd-mailcow    ... done
Pulling acme-mailcow      ... done
Pulling netfilter-mailcow ... done
Pulling watchdog-mailcow  ... done
Pulling dockerapi-mailcow ... done
Pulling solr-mailcow      ... done
Pulling ipv6nat-mailcow   ... done
```

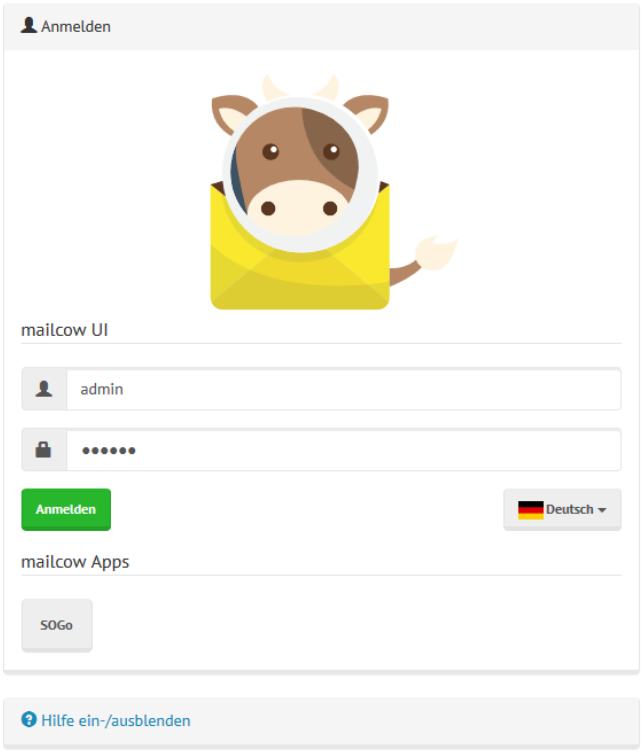
Jetzt können anfangen mit dem Erstellen der Container.

```
docker-compose up -d
```

Um zu sehen was gerade in den Containern passiert, gibt es einen Befehl der alle Logs von den Containern zusammen anzeigt.

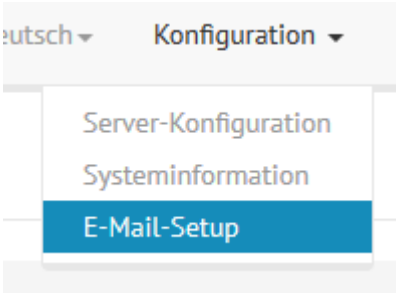
```
docker-compose logs -f
```

Wenn wir nun unseren Browser öffnen und die IP-Adresse des Linux Servers eingeben, dann sollte dieses Interface auftauchen. Der Default User ist **admin** mit dem Passwort **moohoo**

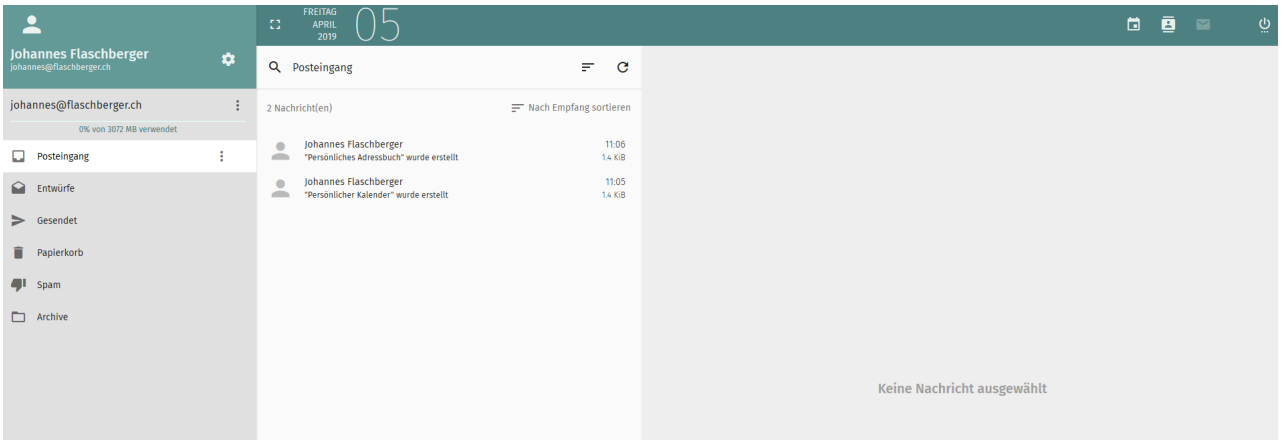


Als nächstes sollte dann das Standard Passwort geändert werden. Dies kann gemacht werden unter **Administrator bearbeiten** > **bearbeiten**.

Um dann die Domains hinzuzufügen und Email-Adressen, gibt es unter dem Reiter **Konfiguration** die Option **E-Mail-Setup**. Dort können dann die Domains und Mailboxen hinzugefügt werden.



Wenn man die Konfiguration abgeschlossen hat, kann man über **Apps** > **SOgo** starten. Dies ist der Webmail Client. Hier ein Beispielbild von dem Webclient:



4.1.1.3 DNS Konfiguration und Portforwarding

Domain	Typ	TTL	Priorität	Wert
flaschberger.ch	SOA	3600	-	ns.hostpoint.ch hostmaster.hostpoint.ch 2018022410 86400 7200 3628800 3600
flaschberger.ch	NS	36000	-	ns.hostpoint.ch
flaschberger.ch	NS	36000	-	ns2.hostpoint.ch
flaschberger.ch	NS	36000	-	ns3.hostpoint.ch
flaschberger.ch	MX	3600	10	mx1.mail.hostpoint.ch
flaschberger.ch	MX	3600	10	mx2.mail.hostpoint.ch
restartit.flaschberger.ch	MX	3600	10	mail.restartit.flaschberger.ch
autoconfig.flaschberger.ch	CNAME	300	-	autoconfig.mail.hostpoint.ch
autoconfig.restartit.flaschberger.ch	CNAME	300	-	mail.restartit.flaschberger.ch
autodiscover.flaschberger.ch	CNAME	300	-	autoconfig-nonssl.mail.hostpoint.ch
autodiscover.restartit.flaschberger.ch	CNAME	300	-	mail.restartit.flaschberger.ch
cloud.restartit.flaschberger.ch	CNAME	300	-	srv-personalcloud.internet-box.ch
cloud.sofie.flaschberger.ch	CNAME	300	-	srv-personalcloud.internet-box.ch
icinga2.pve.flaschberger.ch	CNAME	300	-	srv-personalcloud.internet-box.ch
imap.flaschberger.ch	CNAME	300	-	imap.mail.hostpoint.ch
lists.flaschberger.ch	CNAME	300	-	lists.admin.hostpoint.ch
mail.flaschberger.ch	CNAME	300	-	asmtp.mail.hostpoint.ch
mail2.flaschberger.ch	CNAME	300	-	imap.mail.hostpoint.ch
mc.flaschberger.ch	CNAME	300	-	seafile.internet-box.ch
monitoring.pve.flaschberger.ch	CNAME	300	-	srv-personalcloud.internet-box.ch
personalcloud.flaschberger.ch	CNAME	300	-	srv-personalcloud.internet-box.ch
pop.flaschberger.ch	CNAME	300	-	pop.mail.hostpoint.ch
sgf80q9rf8soo0gouqg7z1pr0fcv8ms8.flaschberger.ch	CNAME	3600	-	s20170307165826.flaschberger.ch
smtp.flaschberger.ch	CNAME	300	-	asmtp.mail.hostpoint.ch
flaschberger.ch	A	300	-	217.26.54.132
*.flaschberger.ch	A	300	-	217.26.54.132
mail.restartit.flaschberger.ch	A	300	-	83.77.54.176
restartit.flaschberger.ch	A	300	-	83.77.54.176
flaschberger.ch	AAAA	300	-	2a00:d70:0:b:2002:0:d91a:3684
*.flaschberger.ch	AAAA	300	-	2a00:d70:0:b:2002:0:d91a:3684
_acme-challenge.flaschberger.ch	TXT	300	-	5sXwisam2hS23yl7dc4ydd6TQQQmFgx_GP2ZU_bC5NY
_acme-challenge.pve.flaschberger.ch	TXT	300	-	ajqwUC454KWByZSeHXj_2-EO3JnjSP6EpAwDB2FRUck
_dmarc.restartit.flaschberger.ch	TXT	300	-	v=DMARC1; p=reject; rua=mailto:mailauth-reports@restartit.flaschberger.ch
dkim._domainkey.restartit.flaschberger.ch	TXT	300	-	v=DKIM1;k=rsa;t=s;s=email;p=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA1fms4oC+aGGQ5p6XtdSVGSkYJTz1j8tl
restartit.flaschberger.ch	TXT	300	-	v=spf1 mx ~all

4.1.1.4 Seafile Installation

Seafile, die Cloud Lösung, hat eine MariaDB integriert. Deshalb brauch es kein zusätzlichen DB Container. Für Seafile sieht das Docker-compose file so aus:

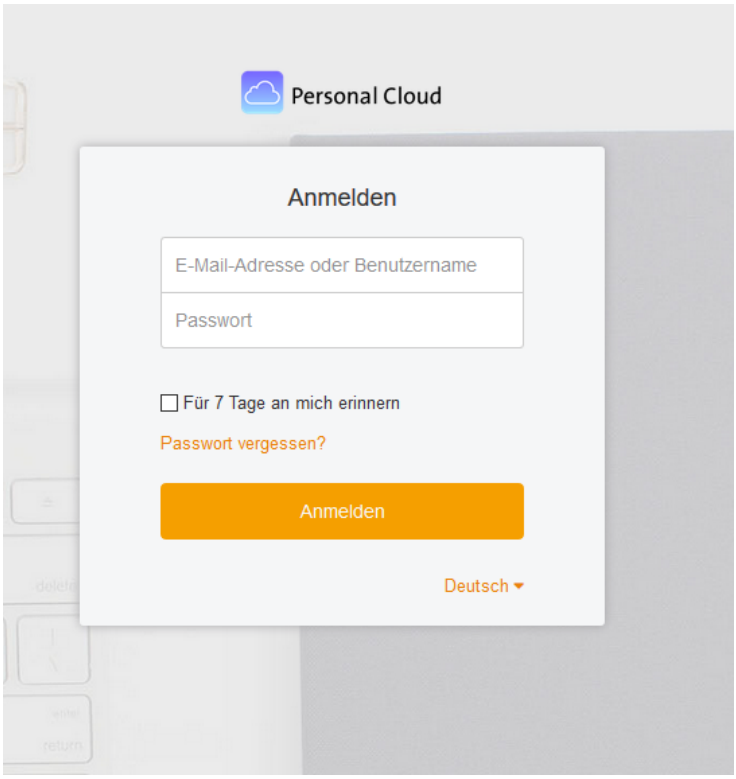
```
version: '3.3'

services:
  seafile:
    image: seafileltd/seafile:6.3.4
```

```
  volumes:
   - /opt/seafile-data:/shared
  ports:
   - "8080:80"
  environment:
   SEAFILE_SERVER_HOSTNAME: seafile.example.com
   SEAFILE_ADMIN_EMAIL: me@example.com
   SEAFILE_ADMIN_PASSWORD: secret
```

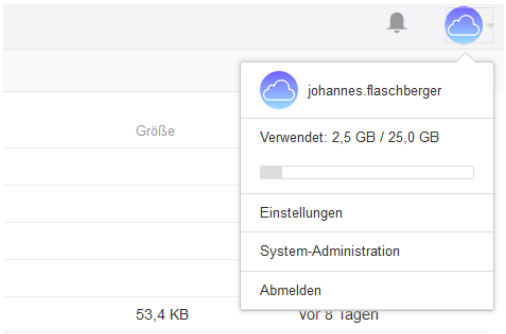
Dann starten wir den Container.

```
docker-compose up
```

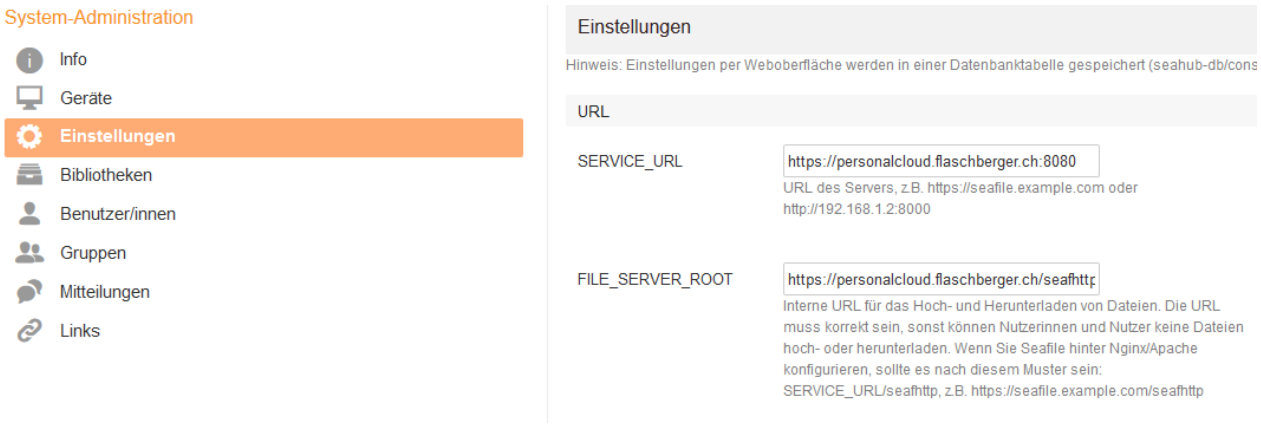
Wenn der Container nun erstellt ist, können wir auf unserem **Browser** die **IP-Adresse** des Linux-Servers eingeben mit dem Port **8080.** Hier können wir uns nun mit dem im docker-compose File festgelegten Admin Account anmelden.



Nach der Anmeldung muss als erstes überprüft werden, ob der Service URL und File Server Root mit der aktuellen Konfiguration übereinstimmt. Hierfür klicken wir oben rechts auf den Account und dann auf System-Administration.



Dann klicken sie auf Einstellungen. Überprüfen sie in den Feldern SERVICE_URL und FILE_SERVER_ROOT, ob die Portangaben stimmen, denn falls nicht, kann nichts hoch- oder runtergeladen werden.



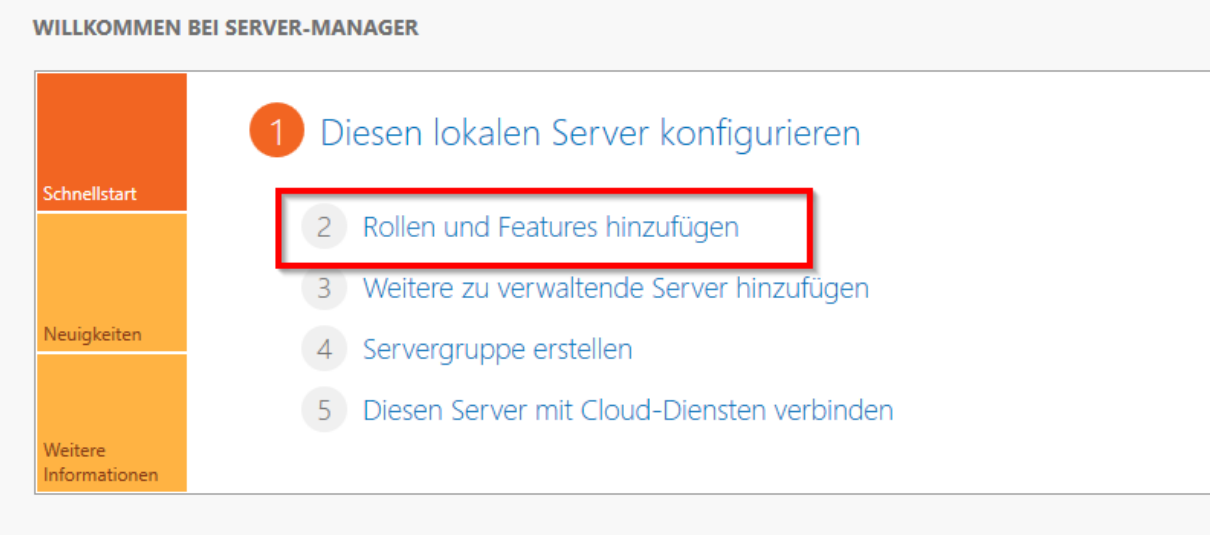
Weitere Vorgehensweise Seafile nach Installation:

1. Benutzergruppen erstellen
2. Benutzer und Shares erstellen
3. Zweifaktor Authentifizierung aktivieren
4. Design und Logos anpassen

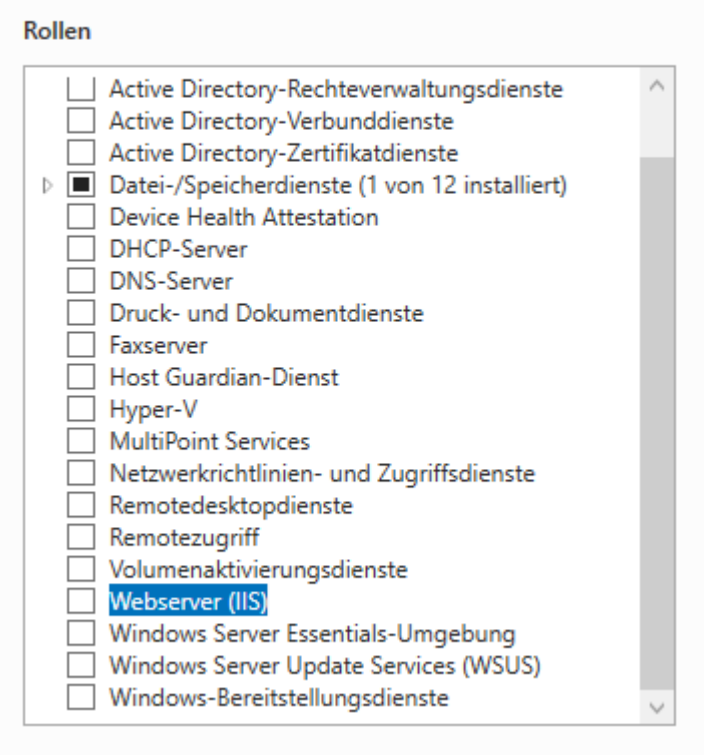
4.1.2 Variante 2 Microsoft IIS

4.1.2.1 Installation Microsoft Internet Information Service

Zuerst müssen die entsprechenden Rollen und Features hinzugefügt werden. Dies kann im Servermanager erledigt werden. Zuerst auf "Rollen & Features hinzufügen:



Danach kann die entsprechende Rolle Als Webserver ausgewählt werden:



4.1.2.2 Installation WordPress

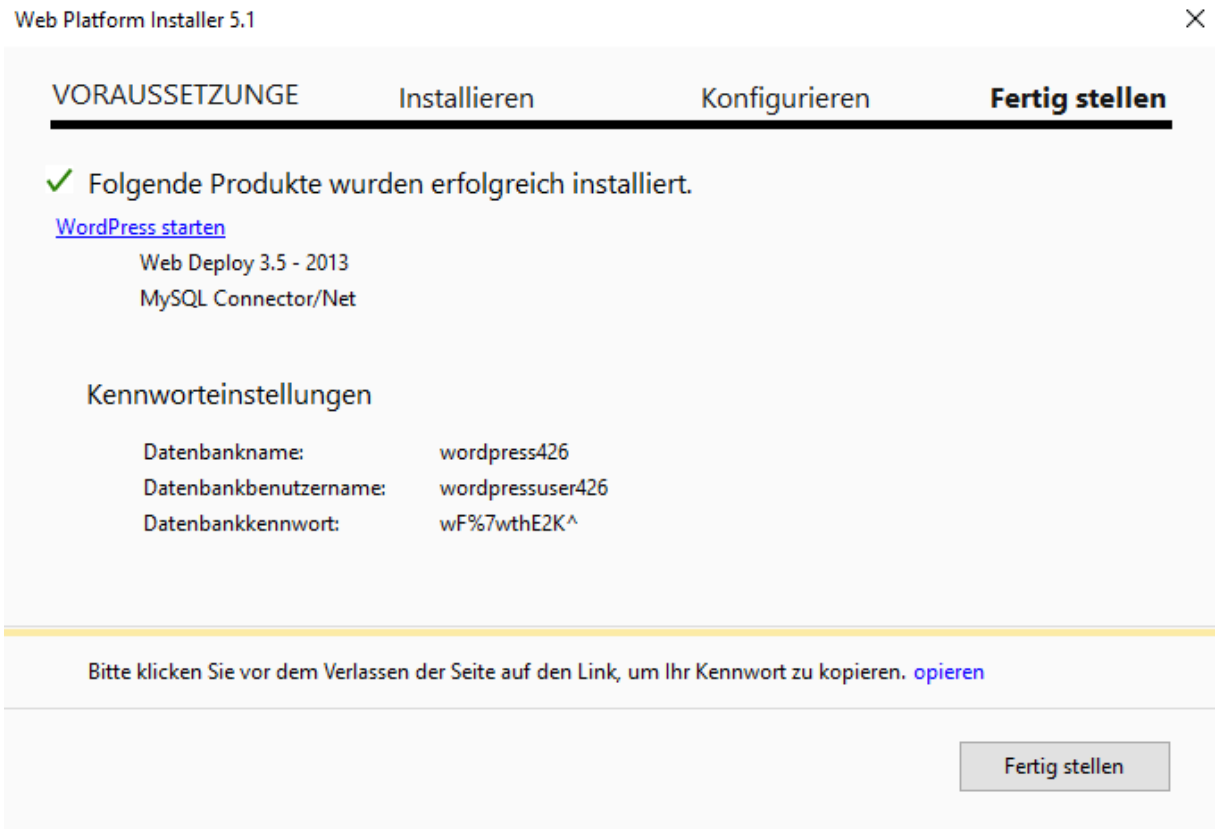
Um WordPress zu installieren muss zuerst von der Microsoft-Webseite der "Microsoft Web Platform Installer" heruntergeladen und installiert werden. Damit kann dann WordPress, sowie alle benötigten Komponenten, wie PHP und MySQL heruntergeladen & installiert werden.

Hier eine Liste der benötigten Komponenten für WordPress, damit dieses überhaupt ordnungsgemäss funktioniert:

✕ **MySQL Windows 5.5 (Englisch)**	Direkter Downloadlink
✕ **PHP 7.3.1 (x64) (Englisch)** Lizenzbedingungen anzeigen	Direkter Downloadlink
✕ **PHP Manager für IIS (Englisch)** Lizenzbedingungen anzeigen	Direkter Downloadlink
✕ **PHP 7.3.1 (x64) For IIS Express (Englisch)** Lizenzbedingungen anzeigen	Direkter Downloadlink
Dateidownloadgröße insgesamt:	**185.22 MB**

Nach der Installation dieser Anwendungen fertiggestellt wurde, kann WordPress auch über den Web Platform Installer installiert werden. Schlussendlich muss noch der "WordPress"-Anwendungsname angegeben werden. Bei uns haben wir hier "restart-it" angegeben.

WICHTIG: Die hier erscheinenden Zugangsdaten müssen sicher irgendwo notiert werden:



4.1.2.3 Konfiguration WordPress im WebGUI

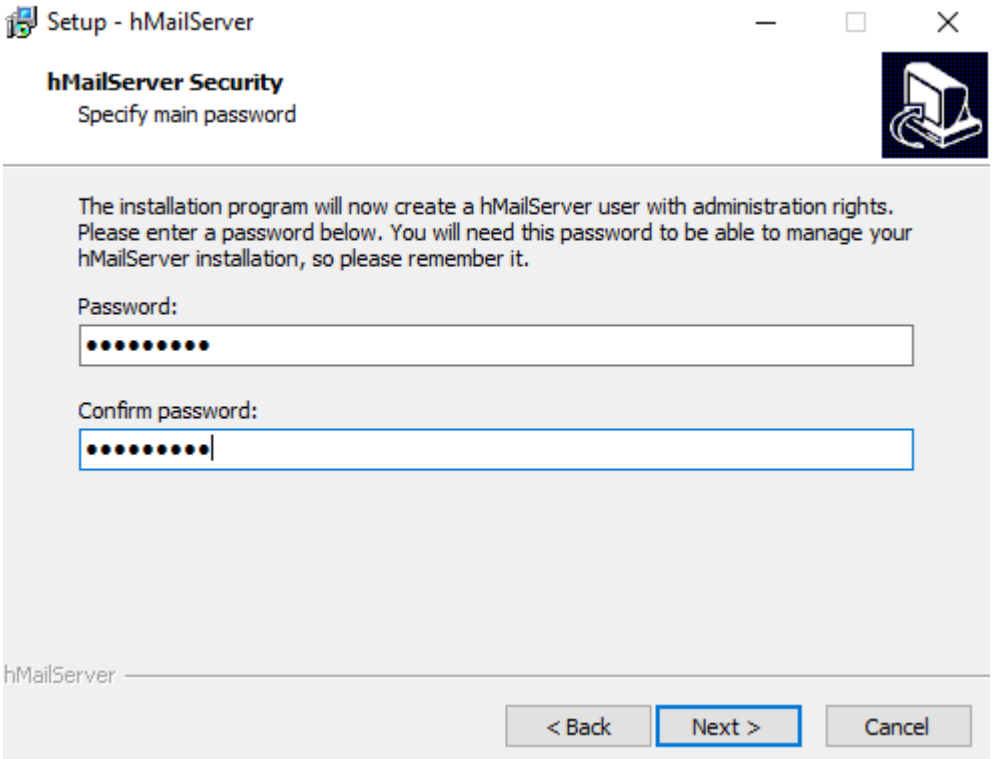
Auf dem gleichen Fenster wie die Zugangsdaten gibt es einen Link, um die erstellte Webseite zu öffnen:

Ist man auf der Webseite angelangt, wird man von WordPress durch das erstmalige Setup durchgeführt.

4.1.2.4 Installation hMailserver

Für hMailServer wird Microsoft .NET-Framework 3.5 benötigt. Dies kann über "Rollen & Features hinzufügen" installiert werden.

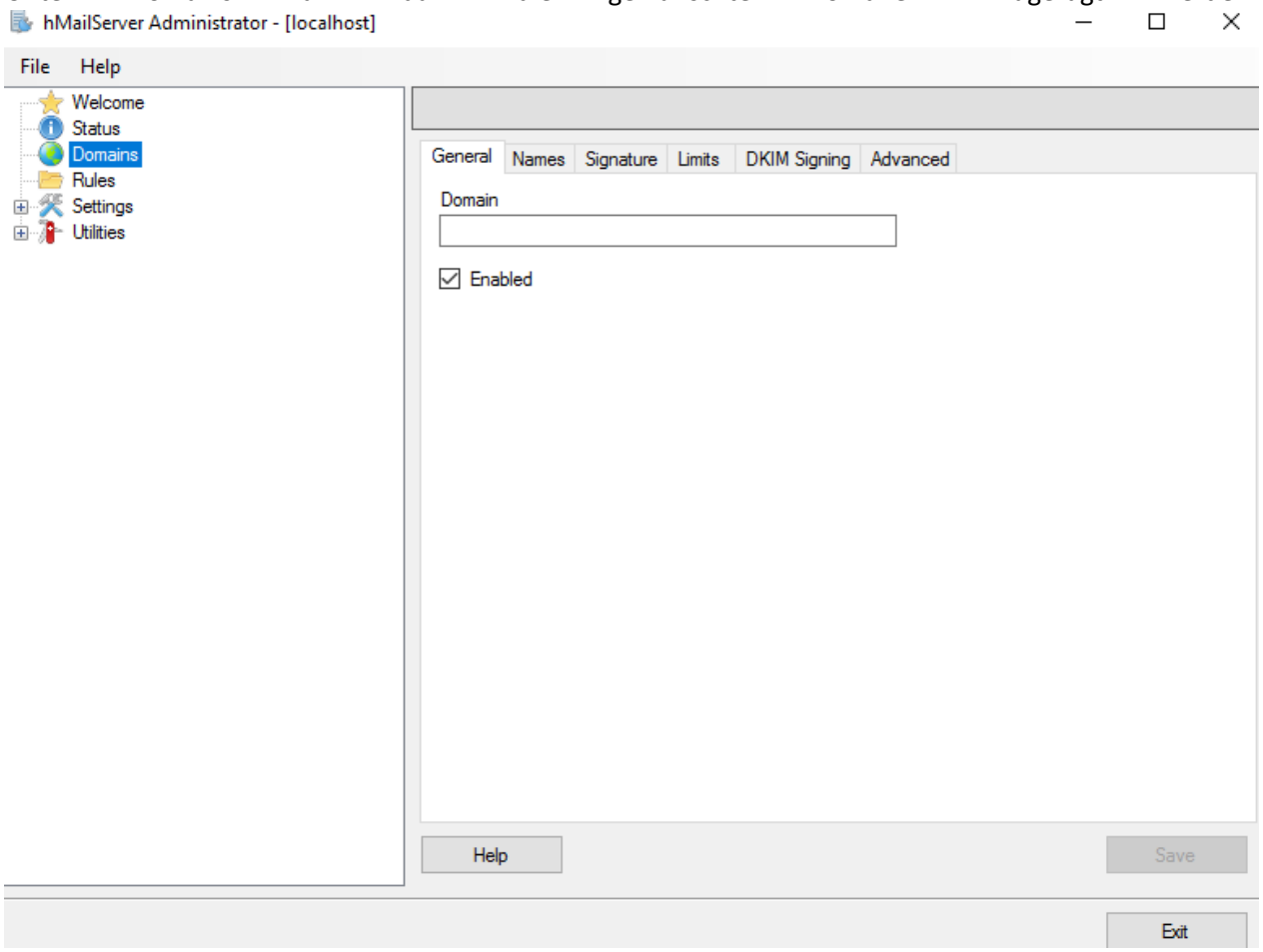
Der hMailServer wird über ein setup.exe installiert. Bei der Installation muss ein sicheres Passwort angegeben werden (für DB):



4.1.2.5 Konfiguration hMailServer

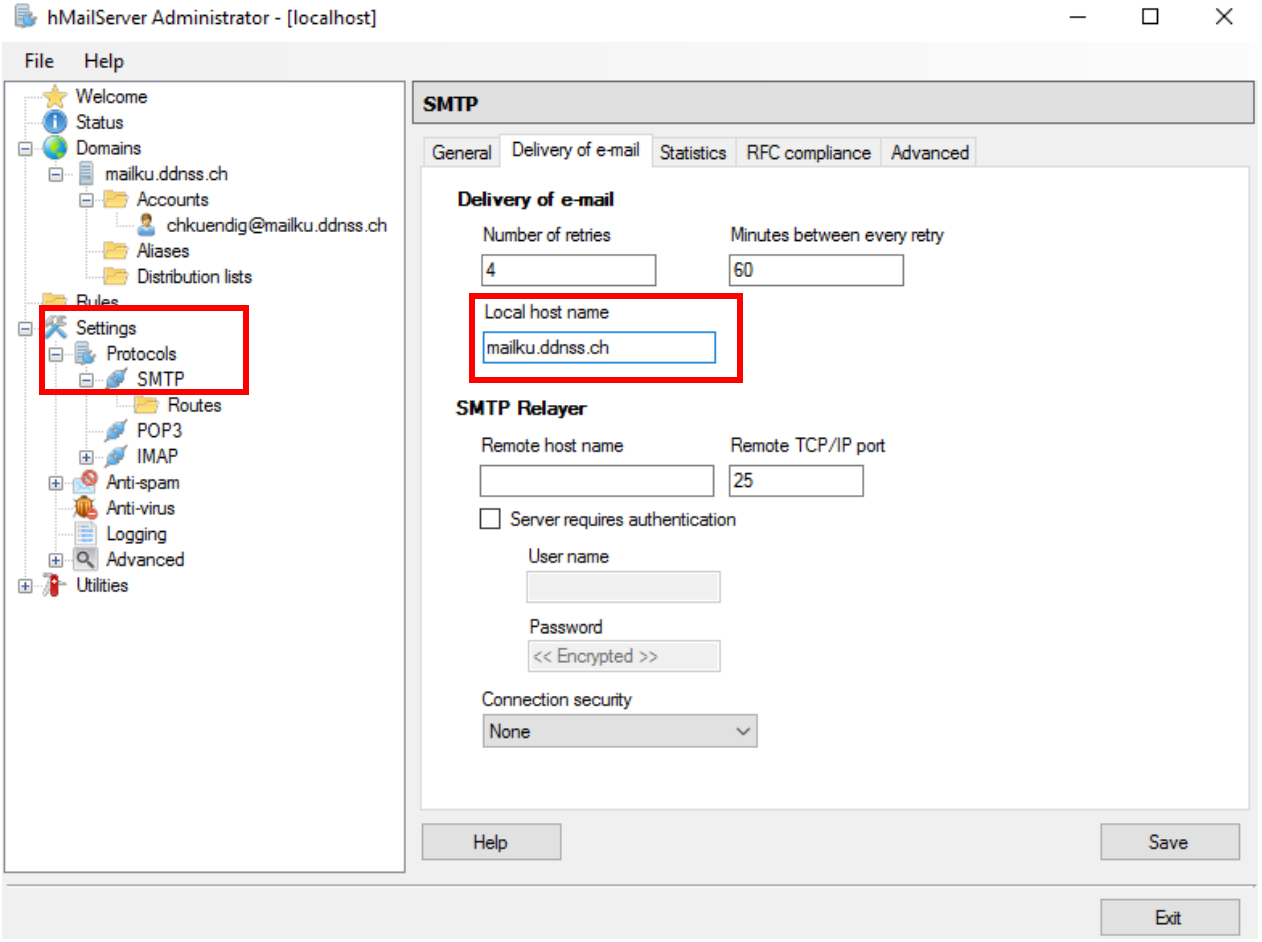
Als erstes im hMailServer Administrator auf localhost klicken -> "Connect". Hier kommt dann eine Passwort-Abfrage. Dies ist das PW, welches im Setup gesetzt wurde.

Unter Domains kann dann die gewünschte Domäne hinzugefügt werden:



Geht man dann weiter auf die Domäne kann unter "Accounts" neue E-Mail-Adressen erstellt werden.

Weiter muss in den Einstellungen von SMTP der FQDN des Servers angegeben werden:



Damit E-Mails auch nach aussen gelangen und auch von aussen empfangen werden können müssen folgende Ports geöffnet werden:

- 25 TCP
- 110 TCP
- 143 TCP
- 587 TCP

4.2 Protokolle

4.2.1 HTTP

HTTP steht für Hypertext Transfer Protokoll und bezeichnet ein zustandloses Protokoll, mit dem sich Daten in einem IP-Netzwerk übertragen lassen. Eine der wichtigsten Anwendungen von HTTP ist, das Übertragen von Webseiten und Daten zwischen einem Webserver und Webbrowser. Das Protokoll wird jedoch nicht nur für diese Anwendung eingesetzt. Das Übertragungsprotokoll WebDAV baut auch auf diesem Protokoll auf. WebDAV wird zu Übertragung von Ordern und Dateien über ein IP-Netz verwendet. Nun wieder zurück zu HTTP. HTTP arbeitet unverschlüsselt auf Layer 7 des ISO/OSI Modell und sendet somit alle Informationen im Klartext. Deshalb ist dieses Protokoll für Anmeldeseiten und Webseiten mit sensiblen Daten nicht geeignet. Für diese Zwecke gibt es eine verschlüsselte Variante von HTTP genannt HTTPS. Das "S" steht für Secure.

Standard Ports:
HTTP = 80
HTTPS = 443

Funktionsweise Beispiel:

```
HEAD / HTTP/1.1
Host: google.ch
User-Agent: curl/7.54.0
Accept: */*

HTTP/1.1 301 Moved Permanently
Location: http://www.google.ch/
Content-Type: text/html; charset=UTF-8
Date: Mon, 15 Apr 2019 08:02:24 GMT
Expires: Wed, 15 May 2019 08:02:24 GMT
Cache-Control: public, max-age=2592000
Server: gws
Content-Length: 218
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
```

Methoden:
GET – Dies ist eine der wichtigsten Methoden des Protokolls HTTP. Diese Methode dient als Anforderung für ein Dokument oder eine Quelle. Die Quelle wird mit Request URL identifiziert.

POST – Mit dieser Methode kann genau das Gegenteil von der GET Methode gemacht werden. Sie übermittelt beispielsweise Formulareingaben an den Webserver.

4.2.2 DNS

DNS ist sozusagen das Telefonbuch des Internets. Es ordnet Namen zu IP-Adressen zu und umgekehrt. Hierfür gibt es verschiedene DNS-Einträge:

- Der "A"-Record ordnet einem Namen eine IP-Adresse zu.
- Der "AAAA"-Record ordnet einem Namen eine IPv6-Adresse zu.
- Der "CNAME"-Record ordnet einem Namen einen anderen Namen zu.
- Der "MX"-Record weist auf einen Mail-Server.
- Der "PTR"-Record ordnet einer IP-Adresse einen Namen zu. Dies ist der Reverse von A / AAAA.
- Im "TXT"-Record kann man einem Namen irgendeinen Text zuordnen.

Jedes Gerät hat mindestens einen DNS-Server hinterlegt. Folgt eine DNS-Abfrage, geht diese zuerst zum hinterlegten DNS-Server. Findet dieser keine Einträge, wird die Anfrage an den nächsten DNS-Server weitergeleitet, bis schlussendlich ein Eintrag gefunden wurde.

4.2.3 POP3

POP3 steht für «Post Office Protocol» und ist ein Client-Server-basiertes E-Mail-Protokoll. Mit POP3 kann der Posteingang abgerufen werden, sowie Mails aus dem Posteingang gelöscht werden. POP3 funktioniert so, dass alle Mails im Posteingang auf dem Webserver heruntergeladen werden, und vom Server gelöscht werden. Das Protokoll dient lediglich zum Abholen, Listen & Löschen von Mails. Zum Senden ist SMTP nötig.

Die Mails werden vom Server über den TCP-Port 110 heruntergeladen. Der Client muss sich gegenüber dem Server mit Benutzernamen und Passwort authentifizieren. Mit verschiedenen Kommandos, welche aus drei bis vier Zeichen bestehen, können dann die Mails heruntergeladen werden. Beim Verbindungsabbau wird das QUIT-Kommando geschickt. Dies verhindert, dass bei einem Verbindungsverlust unvollständig übertragenen Mails verloren gehen, da in diesem Fall kein QUIT-Kommando verschickt wurde.

4.2.4 IMAP

IMAP dient dazu Mails und Ordnerstrukturen vom Mailserver abzurufen und zu bearbeiten. Im Gegensatz zu POP3 wird die abgerufene E-Mail nicht auf dem Server gelöscht, es wird nur eine lokale Kopie davon erstellt.

Die Verbindung wird über den Port TCP 143 aufgebaut. Die Kommunikation funktioniert über Textmeldungen im normalen ASCII-Format. Der Client kann mehrere Befehle senden, ohne eine Antwort des Servers abzuwarten. Nach dem Verbindungsaufbau muss sich der Client authentifizieren. Erst dann ist der Zugriff aus Postfach möglich.

5 Kontrolle (Tests)

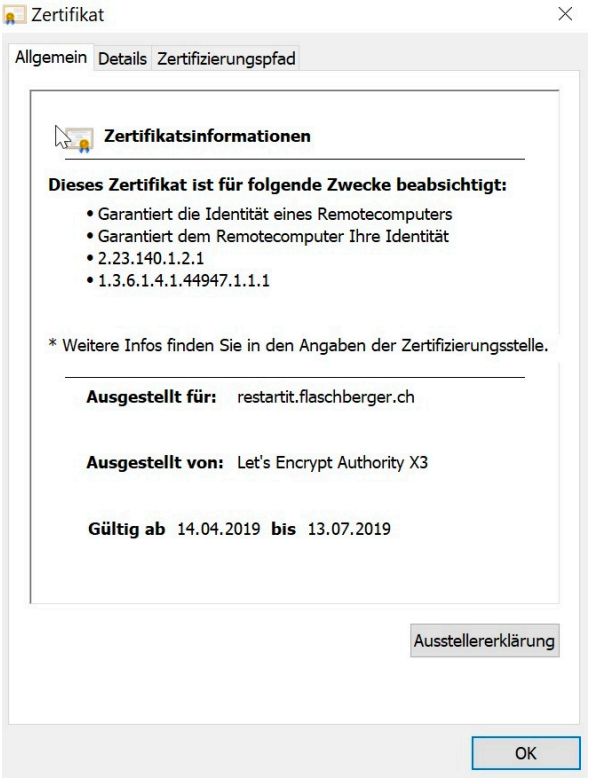
5.1 Testfälle

Service	Was wird getestet?	Soll-Zustand	Ist-Zustand	OK / NOK	Massnahmen
Apache	Webseite über SSL abrufbar	Webseite über sichere Verbindung aufrufbar	Lets Encrypt Zertifikat wird geladen	OK	-
Apache	Webseite ist auf 100 User ausgelegt	Webserver läuft während Test einwandfrei weiter	Mit dem Apache Benchmark tool über 100 Requests pro sekunde ausgeführt	OK	-
Apache	Wordpress für Management läuft	Aktuellste Version von Wordpress installiert & funktionsfähig.	Wordpress Managment kann erreicht werden	OK	-
Apache	Online Chat-Funktion	Auf der Webseite befindet sich ein funktionierender Online-Chat.	Chat funktioniert von beiden Seite	OK	-
Seafile	Cloud nur über SSL	Cloud aufrufbar nur über sichere Verbindung	Lets Encrypt Zertifikat wird geladen	OK	-
Seafile	Gemeinsamer Share	Jeder Benutzer kann im gemeinsamen Share Dokumente ablegen	Auf gemeinsamem Share kann zugegriffen werden	OK	-
Seafile	Persönlicher Speicher	Jeder Benutzer hat zusätzlich noch 100GB persönlicher Speicherplatz	Jeder User hat 100 GB	OK	-
Mailcow	Verdächtige Mails filtern	Verdächtige Mails werden von Mailserver herausgefiltert			
mailcow	Mails versenden & empfangen	E-Mails können versendet und empfangen werden	Mails können versendet und empfangen werden	OK	-
mailcow	Mailserver über sichere Verbindung	E-Mails nur über verschlüsselte Verbindung abrufbar.	Lets Encrypt Zertifikat wird geladen	OK	-

5.1.1 Ergebnisse Tests

5.1.2 Wordpress

5.1.2.1 SSL Zertifikat

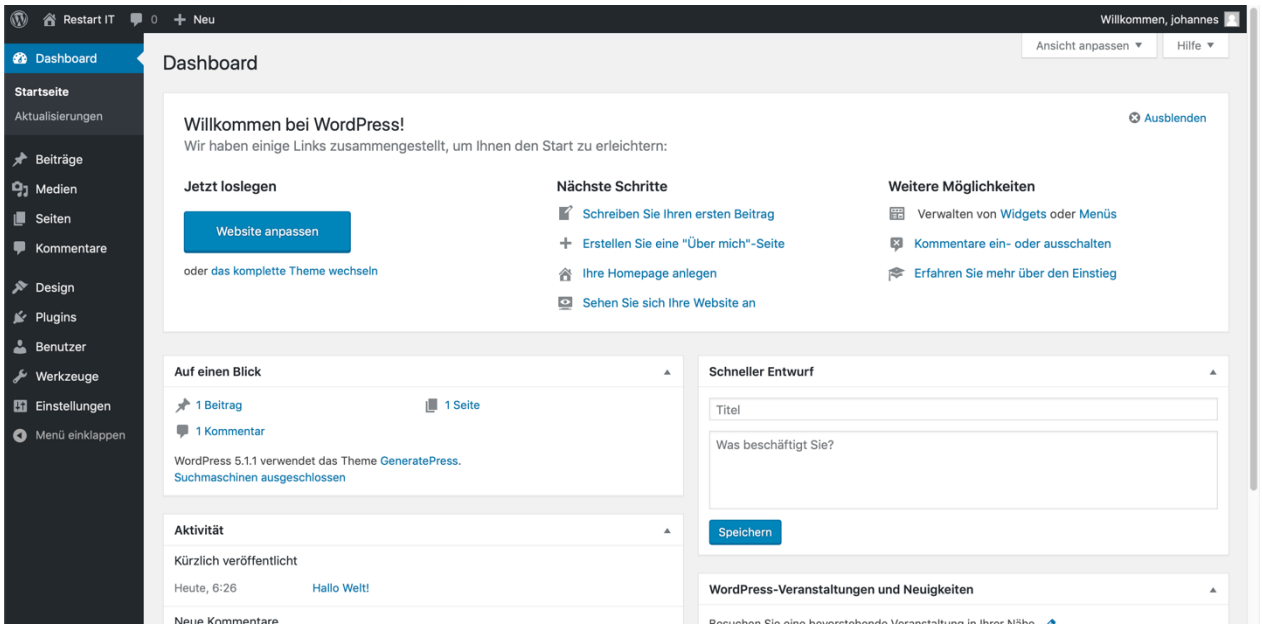


5.1.2.2 Apache Benchmark

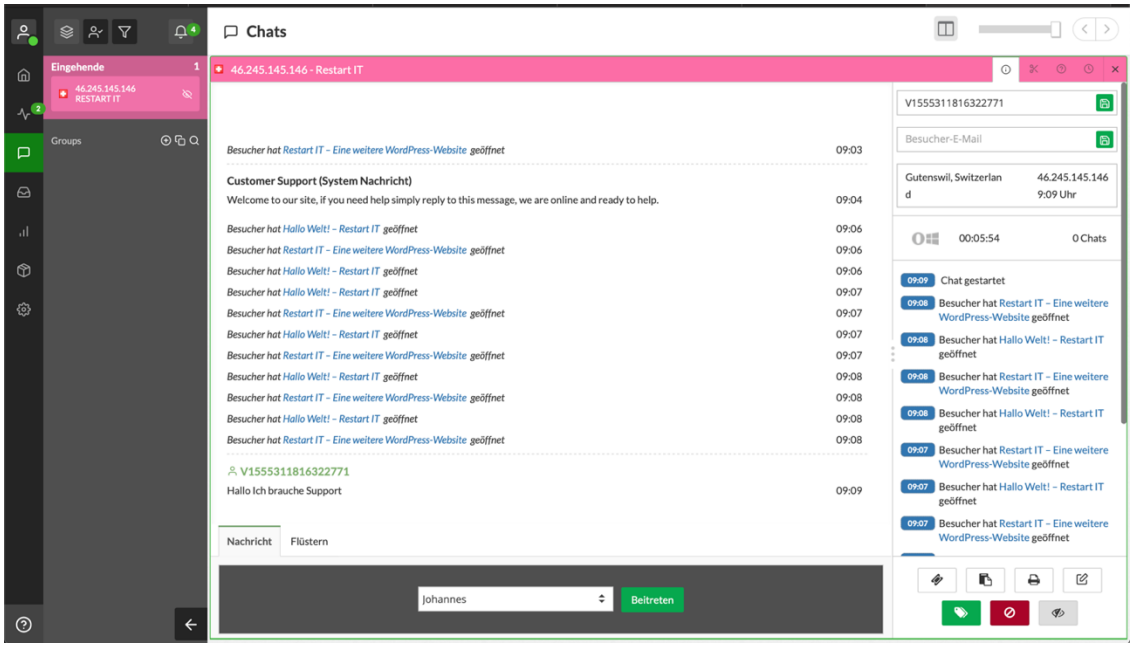
```
Befehl: ab -k -c 1000 -n 1000 172.16.8.106:8080/

Server Software:        Apache/2.4.25
Server Hostname:        172.16.8.106
Server Port:            8080

Document Path:          /
Document Length:        0 bytes

Concurrency Level:      1000
Time taken for tests:   8.566 seconds
Complete requests:      1000
Failed requests:        0
Non-2xx responses:      1000
Keep-Alive requests:    0
Total transferred:      265000 bytes
HTML transferred:       0 bytes
Requests per second:    116.74 [#/sec] (mean)
Time per request:       8565.792 [ms] (mean)
Time per request:       8.566 [ms] (mean, across all concurrent requests)
```

```
Transfer rate:          30.21 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0  557 496.3   1005    1017
Processing:    93 2477 2149.0   1916    7539
Waiting:       92 2476 2149.0   1916    7539
Total:        118 3034 2444.0   2533    8552

Percentage of the requests served within a certain time (ms)
  50%   2533
  66%   3466
  75%   4520
  80%   4787
  90%   8110
  95%   8368
  98%   8504
  99%   8544
 100%   8552 (longest request)
```

5.1.2.3 Management Interface Wordpress

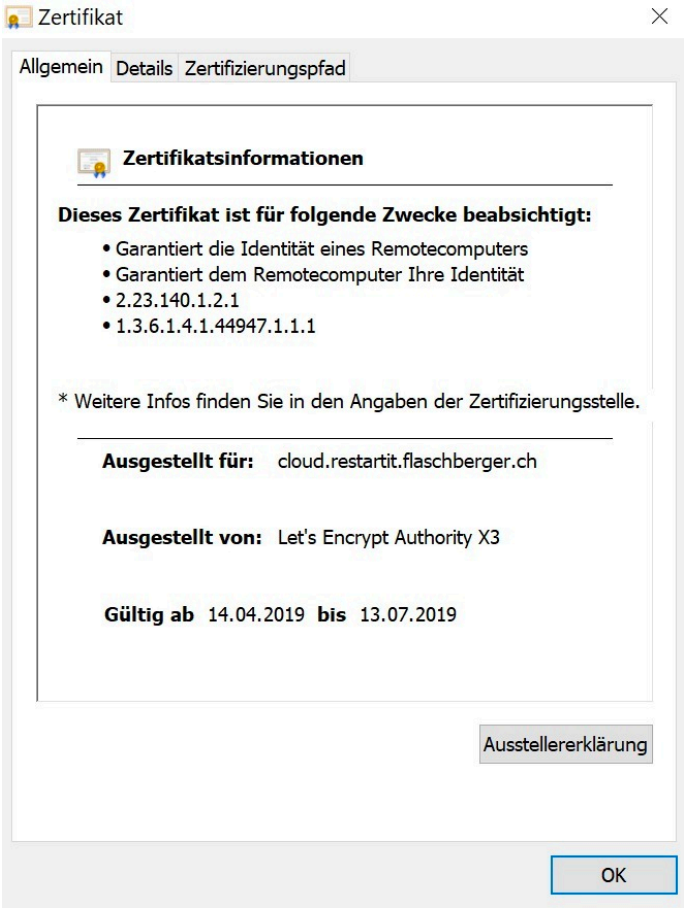


5.1.2.4 Online Chat Integration

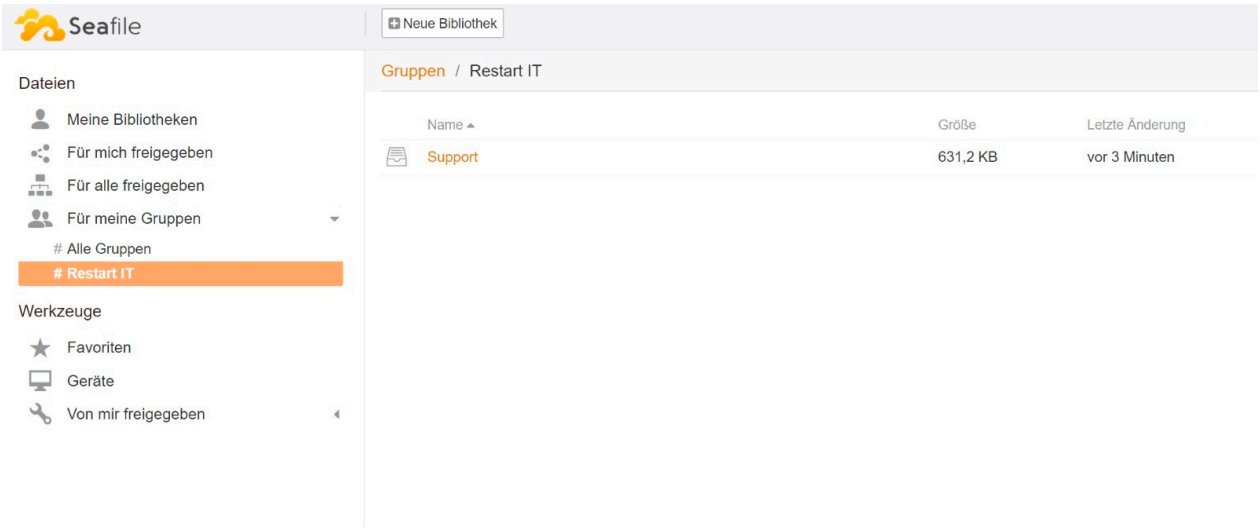


5.1.3 Seafile

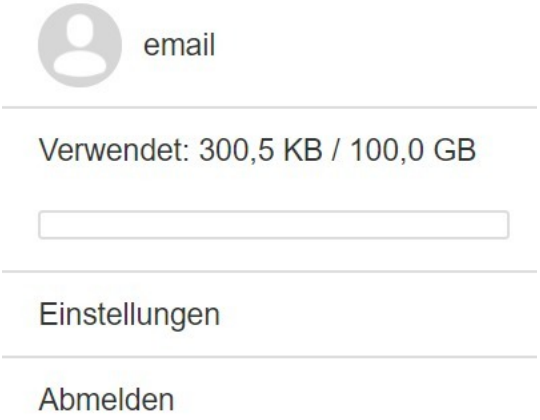
5.1.3.1 SSL Zertifikat



5.1.3.2 Gemeinsamer Share

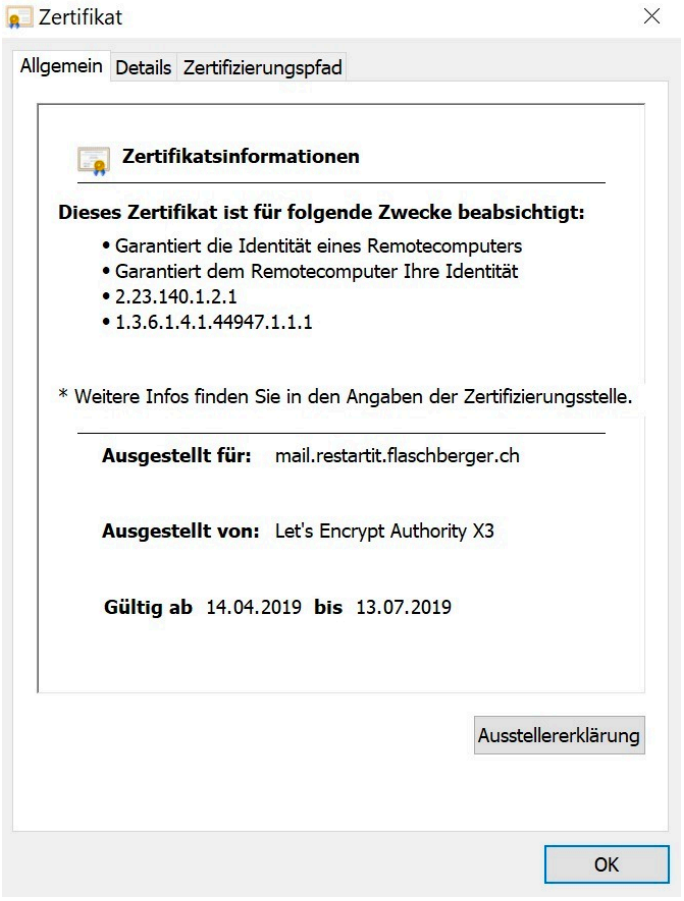


5.1.3.3 Persönlicher Account hat 100 GB

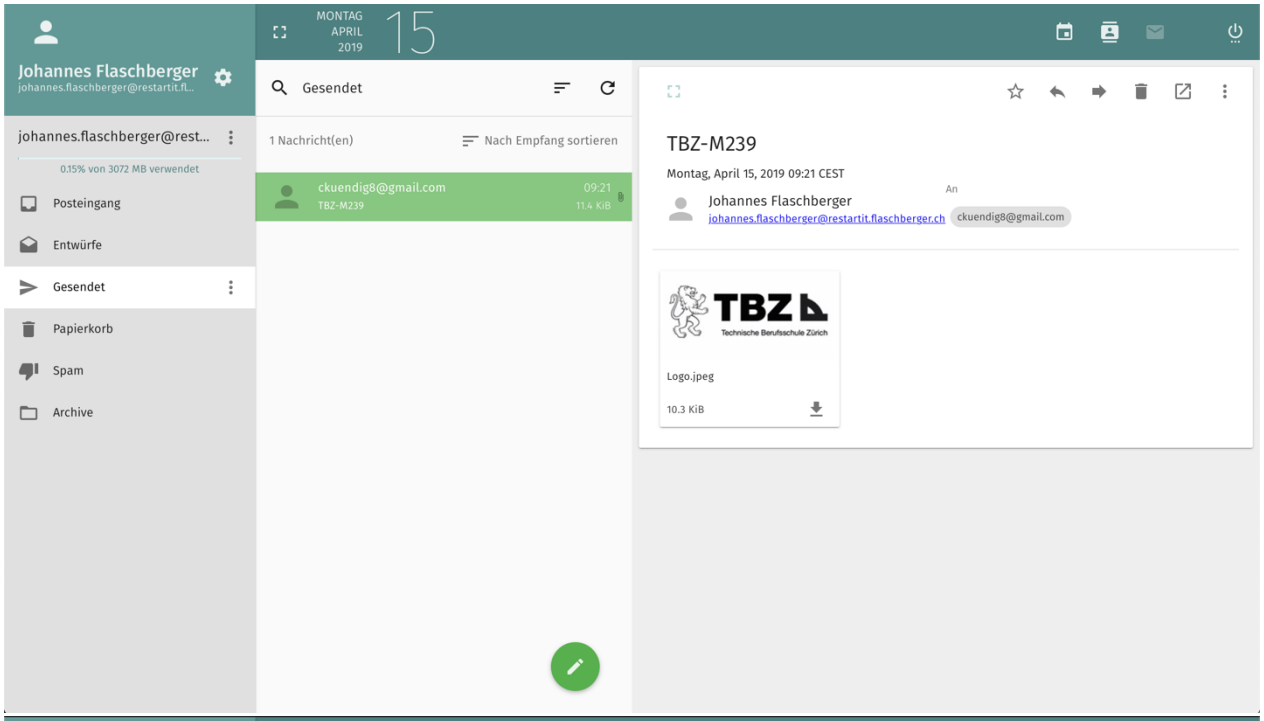


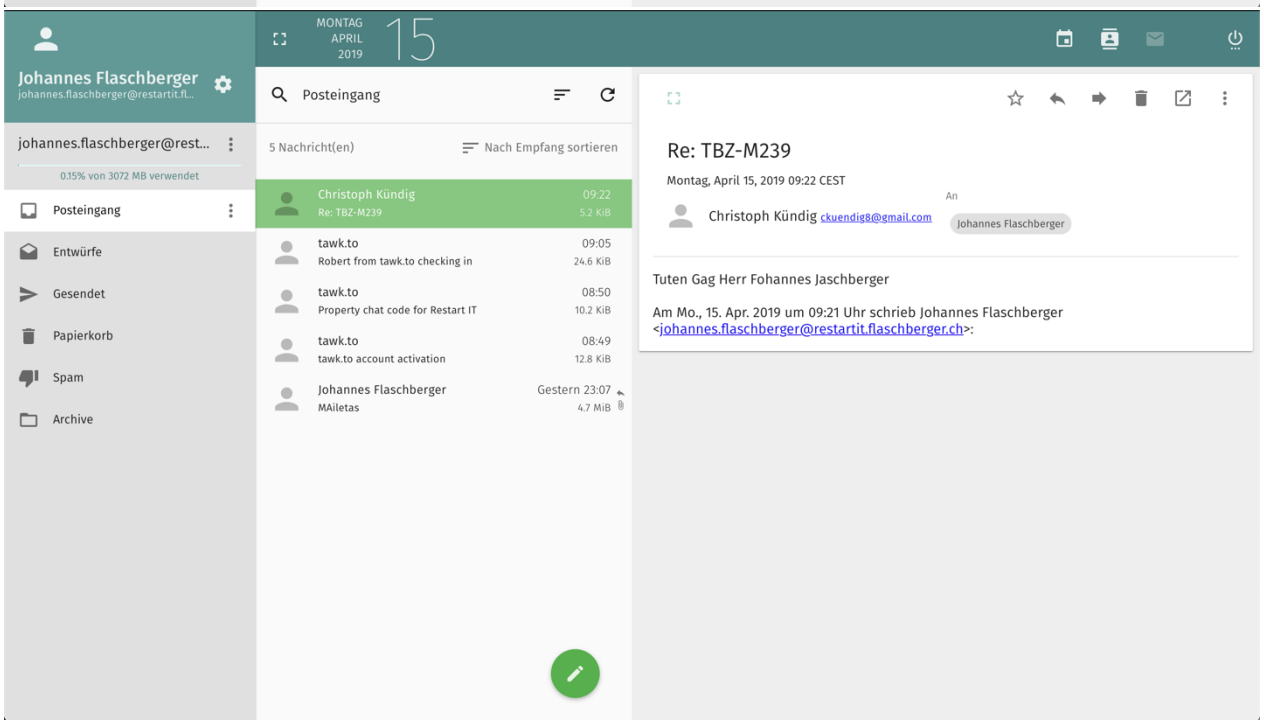
5.1.4 Mailserver

5.1.4.1 Let's encrypt Zertifikat

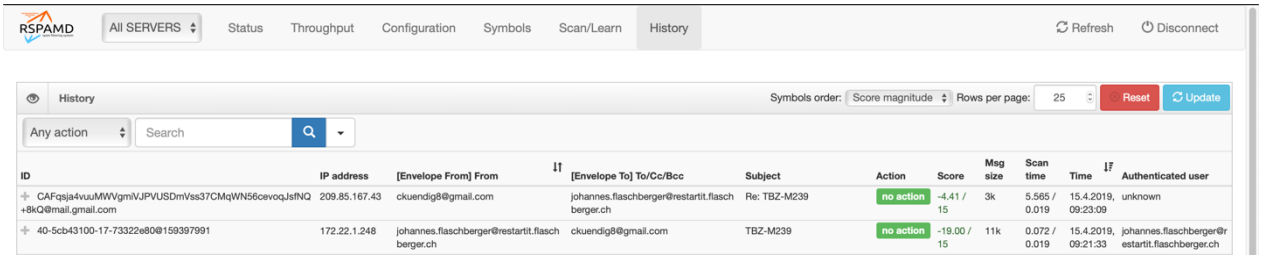


5.1.4.2 Emails versenden und empfangen





5.1.4.3 Mails werden auf verdächtige Inhalte überprüft



6 Auswertung

6.1 Zielerreichung

Wir finden, wir haben unsere Ziele erreicht. Mit der Umsetzung sind wir zufrieden und wir würden es gleich bei einem realen Kunden durchführen.

Bei der Realisation ist uns aufgefallen, das Windows sehr kompliziert und umständlich ist. Mit Linux in Kombination mit Docker hatten wir fast keine Probleme und es ging alles ziemlich Straight Forward. Die DNS Konfiguration war das einzige, dass umständlich war, damit Mails auch über das Internet versendet und empfangen werden.

Unser Zeitplan war hilfreich und wir konnten ihn gut einhalten. Dadurch hatten wir am Ende kein Stress.