Ein Dokuwiki mit Cloud auf dem Raspberry Pi

Fasziniert von dem sehr praktischen Dokuwiki von Markus, habe ich mich auch mal daran gemacht ein eigenes Dokuwiki zu installieren. Ich denke, dass sich das sehr gut im Unterricht verwenden lässt - auch in Anbetracht von BYOD und Distance Learning.
Und wie es so ist, habe ich natürlich auch dafür ein Raspberry Pi (RPi) als Host verwendet...

Für den Austausch von Unterrichtsmaterialien habe ich zudem eine seafile-Instanz als Cloud-Speicher eingerichtet. Diese Kombination von Dokuwiki und seafile scheint mir praktisch.

Das aus dieser Anleitung entstandene Beispiel für ein Dokuwiki ist unter https://alicewiki.ddns.net erreichbar. Die Beispiel-Cloud ist leider offline... 1360

Diese Anleitung dient der Dokumentation, Gedankenstütze und vielleicht auch als Inspiration, ebenso ein Dokuwiki+Seafile-Server im eigenen Hause zu installieren. Bei Anregungen oder Korrekturen gerne Kommentare hinterlassen.

Benötigte Hardware

- Raspberry Pi 3 (die Installation sollte für den RPi 4 aber ebenso funktionieren)
- MicroSD Karte 32GB (evtl. mit Adapter + Kartelesegerät für Computer/Notebook)
- 5V Netzteil (mit min. 2.5A)
- evtl. Ethernet-Kabel
- evtl. Gehäuse für Raspberry Pi

- Computer/Notebook
- Internet (mit Zugang zum Router)

Raspberry Pi OS installieren

Zuerst muss die microSD-Karte mit dem Betriebssystem *Raspberry Pi OS* bespielt werden. Dazu wird die microSD-Karte direkt oder mit einem Kartenlesegerät mit dem PC verbunden und formatiert.
Nun kann mittels dem Raspberry Pi Imager Raspberry Pi OS installiert werden:

1. Imager installieren.
2. Imager starten.
3. Bei „Choose OS“ → „Raspberry Pi OS (other)“ → „Raspberry Pi OS Lite (32-Bit)“ auswählen.
4. Bei „Choose SD“ die verwendete microSD-Karte auswählen.
5. Mit „Write“ den Installationsprozess starten.
6. Nach Abschluss auf „Continue“ klicken, den Imager schliessen und die microSD-Karte vom PC trennen.

Damit wir keinen Monitor, Maus und Tastatur für den RPi benötigen, müssen wir noch *Secure Shell* - kurz SSH - (Protokoll zur Kommunikation übers Netzwerk) freischalten. Die microSD-Karte nochmals mit dem PC verbinden und im Laufwerk mit dem Namen „boot“ eine neue leere Textdatei erstellen. Die neue Datei umbenennen in `ssh`. Wichtig: ohne Dateiendung `.txt`. In Windows 10 können im Datei Explorer Dateiendungen mittels dem Reiter „Ansicht“ → „Ein-/ausblenden“ → „Dateinamenerweiterungen“ angezeigt werden.

Falls WLAN als Internetverbindung für den RPi verwendet werden soll, muss zusätzlich eine Datei `wpa_supplicant.conf` mit folgendem Inhalt ebenfalls im Laufwerk „boot“ der microSD-Karte hinzugefügt werden.

```
country=ch
update_config=1
ctrl_interface=/var/run/wpa_supplicant

network={
 scan_ssid=1
 ssid="NameDeinesWLAN"
 psk="DeinWLANPasswort"
}
```

Nun kann die microSD-Karte in den RPi eingesetzt und dieser mit dem Netzteil mit Strom versorgt werden. Falls LAN für die Internetverbindung verwendet werden soll, ist natürlich noch der RPi mit dem Router per Ethernetkabel zu verbinden.

SSH-Verbindung mit dem RPi

Um sich mit dem RPi per SSH zu verbinden, benötigen wir einen SSH-Client. Bei Linux, Windows 10 und MacOS ist dieser bereits integriert. Für ältere Windows-Versionen eignet sich PuTTY als SSH-Client sehr gut.

Der RPi besitzt eine IP-Adresse unter welcher wir ihn im Netzwerk erreichen. Diese IP-Adresse findet sich am einfachsten in den Einstellungen des Routers: Im Browser die Adresse `192.168.1.1` eingeben (oder je nach Router: `192.168.178.1`, `192.168.1.2`, `192.168.2.1`, Fritz!Box: `fritz.box`) und in den Routereinstellungen unter „Heimnetz“ oder „Netzwerk“ die IP-Adresse des RPi auslesen. Der RPi sollte dort als Gerät mit dem Namen `raspberrypi` mit entsprechender IP aufgeführt sein.

Nun können wir uns per SSH mit dem RPi verbinden (für Windows Versionen älter als Windows 10 siehe SSH-Windows) :

1. Terminal öffnen.
2. `ssh pi@192.168.178.20` eintippen und Enter drücken (IP-Adresse anpassen).
3. Es wird ein Passwort verlangt. Dieses lautet: `raspberry`
4. Authentifizierung mit `yes` bestätigen.
5. Die Verbindung mit dem RPi ist nun aktiv und im Terminal sollte `pi@raspberrypi:~ $` zu sehen sein.

Raspberry Pi OS einrichten

Bevor wir Dokuwiki und seafile installieren, nehmen wir noch folgende Einstellungen in Raspberry Pi OS vor.

Upgraden

1. Zum Upgraden des Betriebsystems `sudo apt-get update && sudo apt-get upgrade` eingeben und mit Enter bestätigen.
2. Nachfrage zur Installation der Upgrades mit Y und Enter bestätigen.
3. Das Upgraden dauert eine Weile. Nach Abschluss erscheint wieder der Prompt `pi@raspberrypi:~ $`.

Passwort, Benutzer- und Hostname ändern

Das Standartpasswort `raspberry` sollten wir zur Sicherheit ändern.

1. `sudo raspi-config` im Terminal eingeben und mit Enter bestätigen.
2. Die 1. Option „Change User Password“ auswählen und das Passwort durch zweimalige Eingabe ändern. Ein sicheres Passwort ist Pflicht, falls der RPi übers Internet erreicht werden soll – lieber zu kompliziert als zu einfach 😉.
3. Der Hostname mittels `sudo raspi-config` unter „Network Options“ → „Hostname“ ändern (z.B. zu `dokuwikiserver`).
4. Die Konfiguration zuerst mit <Back> und dann mit <Finish> beenden.
5. Die Nachfrage zum Reboot mit <Yes> bestätigen.

Nun nochmals mit SSH mit dem RPi verbinden, jedoch das neue Passwort verwenden.

1. Mittels Befehl `sudo adduser alice` einen neuen Benutzernamen anlegen (Alice kann natürlich beliebig ersetzt werden…).
2. Zweimal ein Passwort eingeben.
3. Falls gewünscht Zusatzangaben zur Person machen (können durch Enter aber auch übersprungen werden).
4. Mit Y bestätigen.
5. Damit der neue Name auch alle Berechtigungen hat, müssen wir diese mit dem Befehl `sudo usermod -a -G adm,dialout,cdrom,sudo,audio,video,plugdev,games,users,input,netdev,gpio,i2c,spi alice` erteilen.
6. Mit `sudo su - alice` zum neuen Namen wechseln.
7. Alle Prozesse mit dem Benutzer `pi` mittels `sudo pkill -u pi` beenden.
8. Nun wird die SSH Verbindung gestoppt.
9. Erneut per SSH mit dem RPi verbinden, nun jedoch mit `ssh alice@192.168.178.20` (IP-Adresse anpassen).
10. Mit `sudo deluser -remove-home pi` den Benutzer `pi` löschen.

Zeitzone und Sprache ändern

Mittels `sudo raspi-config` können unter „Localisation Options“ die Zeitzone, die Sprache und das Keyboardlayout geändert werden. Am Besten wird nach der Änderung der RPi neugestartet.

Dokuwiki-Server erstellen

Webserver-Software nginx + php installieren

Nun können wir die Dokuwiki-Installation durchführen. Wir müssen uns dafür per SSH mit dem RPi verbinden und zuerst die Webserver-Software `nginx` und `php` installieren.

1. Die Paketliste mit `sudo apt-get update` aktualisieren.
2. Die Installation von `nginx` mit `sudo apt-get install nginx -y` starten.
3. Die Installation von `php` mit `sudo apt-get install php php-fpm php-mbstring php-xml php-gd php-sqlite3 php7.4-sqlite -y` durchführen.

Um zu testen, ob die Installationen geklappt haben, prüfen wir diese für `nginx` mit dem Befehl `systemctl status nginx.service`. Die Ausgabe im Terminal sollte etwa so aussehen:

```
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor
preset: en
   Active: active (running) since Fri 2020-05-15 01:24:19 CEST; 11h ago
     Docs: man:nginx(8)
 Main PID: 494 (nginx)
    Tasks: 5 (limit: 2200)
   Memory: 11.7M
   CGroup: /system.slice/nginx.service
           ├─494 nginx: master process /usr/sbin/nginx -g daemon on;
master_proc
           ├─495 nginx: worker process
           ├─496 nginx: worker process
           ├─497 nginx: worker process
           └─498 nginx: worker process

May 15 01:24:18 dokuwikiserver systemd[1]: Starting A high performance web
serve
May 15 01:24:19 dokuwikiserver systemd[1]: Started A high performance web
server
```

Für `php` verwenden wir den Befehl `systemctl status php7.4-fpm.service`. Im Terminal sollte die Ausgabe in etwa so aussehen:

```
● php7.4-fpm.service - The PHP 7.4 FastCGI Process Manager
   Loaded: loaded (/lib/systemd/system/php7.4-fpm.service; enabled; vendor
prese
   Active: active (running) since Fri 2020-05-15 12:28:16 CEST; 2min 32s ago
```

```
     Docs: man:php-fpm7.3(8)
 Main PID: 9162 (php-fpm7.3)
   Status: "Processes active: 0, idle: 2, Requests: 0, slow: 0, Traffic:
0req/se
    Tasks: 3 (limit: 2200)
   Memory: 5.7M
   CGroup: /system.slice/php7.4-fpm.service
           ├─9162 php-fpm: master process (/etc/php/7.4/fpm/php-fpm.conf)
           ├─9163 php-fpm: pool www
           └─9164 php-fpm: pool www

May 15 12:28:15 dokuwikiserver systemd[1]: Starting The PHP 7.4 FastCGI
Process
May 15 12:28:16 dokuwikiserver systemd[1]: Started The PHP 7.4 FastCGI
Process M
```

Dokuwiki installieren

Nun haben wir die benötigte Software, um Dokuwiki zu installieren.

1. Mit `cd /var/www` ins Verzeichnis der Webserver wechseln.
2. Die stabile Dokuwiki-Version mittels `sudo wget` https://download.dokuwiki.org/src/dokuwiki/dokuwiki-stable.tgz herunterladen.
3. Mit `sudo tar xfz dokuwiki-stable.tgz` das Paket entpacken.
4. Danach das Paket mit `sudo rm dokuwiki-stable.tgz` löschen.
5. Der Einfachheit halber den Dokuwiki-Ordner unbenennen: `sudo mv /var/www/dokuwiki-2020-07-29 /var/www/dokuwiki` .
6. Die Berechtigungen für den Webserver-User mit `sudo chown -R www-data:www-data /var/www/dokuwiki` setzen.

nginx einrichten

Damit wir auf das Dokuwiki zugreifen können, muss noch `nginx` entsprechend konfiguriert werden. Im Moment wird Dokuwiki nur im lokalen Netzwerk erreichbar sein. Wir können eine beliebige Domain wählen z.B.: `dokuwikizuhause`. Mit dem Befehl: `sudo nano /etc/nginx/sites-enabled/dokuwikizuhause.conf` erstellen wir die nötige Konfigurationsdatei. In diese fügen wir folgenden Code ein:

```
server {
    listen              80;
    listen              [::]:80;
    server_name         dokuwikizuhause;
    root /var/www/dokuwiki;

    location / {
        index doku.php;
        try_files $uri $uri/ @dokuwiki;
    }
```

```
    location @dokuwiki {
        rewrite ^/_media/(.*) /lib/exe/fetch.php?media=$1 last;
        rewrite ^/_detail/(.*) /lib/exe/detail.php?media=$1 last;
        rewrite ^/_export/([^/]+)/(.*) /doku.php?do=export_$1&id=$2 last;
        rewrite ^/(.*) /doku.php?id=$1&$args last;
    }
    location ~ \.php$ {
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_pass unix:/run/php/php7.4-fpm.sock;
    }

    location ~ /(data|conf|bin|inc)/ {
        deny all;
    }
}
```

Einfügen lässt sich der Code durch Rechts-klicken auf das Terminal → „Einfügen“. Danach die Änderungen mit CTRL+S speichern (evtl. andere Tastenkombination bei MacOS oder Windows z.B. CTRL+0) und die Datei mit CTRL+X schliessen.
Den nginx-Server starten wir mit `sudo systemctl restart nginx.service` neu.

Dokuwiki im Netzwerk erreichen

Damit wir die Domain `dokuwikizuhause` verwenden können, müssen wir auf unserem PC noch die Hosts-Datei anpassen (für MacOS bzw. für Windows). Die Hosts-Datei öffnen und die Zeile: `192.168.178.20 dokuwikizuhause` hinzufügen (IP-Adresse anpassen).

Dokuwiki ist nun im Browser unter der Adresse `http://dokuwikizuhause` in eurem Netzwerk erreichbar. Damit wir Dokuwiki fertig einrichten können, gehen wir zu `http://dokuwikizuhause/install.php` und installieren Dokuwiki nach unseren Wünschen. Nach Abschliessen durch `speichern` ist die Installation von Dokuwiki abgeschlossen. Wir können also das Installationsskript auf dem RPi löschen. Dazu per SSH verbinden und den Befehl `sudo rm /var/www/dokuwiki/install.php` eingeben.

Das wars! Nun ist Dokuwiki installiert und im eigenen Netzwerk unter `http://dokuwikizuhause` zu erreichen, kann bearbeitet und gefüllt werden…
Um das Dokuwiki auch aus dem Internet zugänglich zu machen siehe Internetzugang.

seafile-Server erstellen

Seafile ist ein OpenSource-Fileshare-Server. Zwar hat seafile nicht soviele Erweiterungsmöglichkeiten wie beispielsweise nextcloud, aber dadurch ist die Performance auf dem RPi auch etwas besser.

seafile installieren

Für die Installation müssen wir uns wieder per SSH mit dem RPi verbinden.

1. Zuerst die Paketliste aktualisieren mit `sudo apt-get update`.
2. Und die benötigten Pakete für die seafile-Installation installieren: `sudo apt-get install python3 python3-setuptools python3-pip sqlite3 -y`
3. Ebenso: `sudo pip3 install pillow pycryptodome==3.12.0 cffi==1.14.0`
4. Für die Installation einen Benutzer `seafile` mit dem Befehl `sudo useradd -m -p seafile -s /bin/bash seafile` anlegen.
5. Zu diesem Benutzer mit `sudo su seafile` wechseln.
6. Mit `cd` ins Hauptverzeichnis des neuen Benutzers wechseln.
7. seafile herunterladen mit `wget https://github.com/haiwen/seafile-rpi/releases/download/v9.0.2/seafile-server-9.0.2-bullseye-arm32v7l.tar.gz`
8. (Für Raspberry Pi 4 die Version `seafile-server-9.0.2-bullseye-arm64v8l.tar.gz` verwenden)
9. Die Datei mit `tar xfz seafile-server-9.0.2-bullseye-arm32v7l.tar.gz` entpacken.
10. Das heruntergeladene Paket mit `rm seafile-server-9.0.2-bullseye-arm32v7l.tar.gz` löschen.
11. In den seafile-Ordner wechseln: `cd seafile-server-9.0.2`
12. Die Installation mit `./setup-seafile.sh` starten.
13. Dem Skript folgen:
    1. Servername z.B. `alicecloud` wählen
    2. Bei der `domain/IP` die IP-Adresse des RPi eingeben `192.168.178.20` (IP-Adresse anpassen).
    3. Das Verzeichnis und der Port `8082` mit Enter als `default` bestätigen.
    4. Enter drücken für die seafile-Installation.
    5. Zum Abschluss für die Einrichtung von seahub nochmals Enter drücken.

Wir wechseln nun ins Hauptverzeichnis mit `cd` und danach in den Seafile-Installationsordner `cd seafile-server-latest`. Nun können wir seafile mit dem Befehl `./seafile.sh start` starten. Zusätzlich müssen wir noch seahub mit dem Befehl `./seahub.sh start` starten und das Administrator-Login einrichten (diese Daten benötigen wir später fürs Login in seafile). Wir stoppen seafile und seahub für die weitere Konfiguartion wieder mit `./seafile.sh stop` und `./seahub.sh stop`.

Autostart einrichten

Damit seafile automatisch gestartet wird, richten wir einen Autostart ein. Zuerst mit `exit` zum Hauptbenutzer wechseln (in diesem Fall `alice`). Den Befehl `sudo nano /etc/systemd/system/seafile.service` ausführen und in den Editor mit Rechtsklick den

folgenden Code einfügen:

```
[Unit]
Description=Seafile
# add mysql.service or postgresql.service depending on your database to the
line below
After=network.target

[Service]
Type=forking
ExecStart=/home/seafile/seafile-server-latest/seafile.sh start
ExecStop=/home/seafile/seafile-server-latest/seafile.sh stop
User=seafile
Group=seafile

[Install]
WantedBy=multi-user.target
```

Mit CTRL+S und CTRL+X die Änderungen speichern und den Editor schliessen.

Dasselbe noch für seahub einrichten. Mit dem Befehl `sudo nano /etc/systemd/system/seahub.service` die Konfigurationsdatei öffnen und diesen Code einfügen:

```
[Unit]
Description=Seafile hub
After=network.target seafile.service

[Service]
Environment="LC_ALL=C"
Type=forking
# change start to start-fastcgi if you want to run fastcgi
ExecStart=/home/seafile/seafile-server-latest/seahub.sh start
ExecStop=/home/seafile/seafile-server-latest/seahub.sh stop
User=seafile
Group=seafile

[Install]
WantedBy=multi-user.target
```

Den Autostart mit `sudo systemctl enable seafile seahub` aktivieren. Nun können wir seafile auch als `alice` (root-User) steuern mit `sudo systemctl start seafile seahub` zum Starten, `sudo systemctl stop seafile seahub` zum Stoppen und `sudo systemctl restart seafile seahub` zum Neustarten.

nginx einrichten

Damit der seafile-Server auch im Netzwerk erreicht werden kann, müssen wir `nginx` noch entsprechend einrichten. Falls noch nicht installiert, kann dies mit `sudo apt-get install nginx` nachgeholt werden.

Dann erstellen wir die Konfigurationsdatei für seafile mit `sudo nano /etc/nginx/sites-enabled/seafile.conf` und fügen folgenden Code ein (der Servername `alicecloud` nach Wunsch ändern):

```
server {
    listen 80;
    listen [::]:80;
    server_name alicecloud;

    proxy_set_header X-Forwarded-For $remote_addr;

    location / {
         proxy_pass         http://127.0.0.1:8000;
         proxy_set_header   Host $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-Host $server_name;
         proxy_read_timeout  1200s;


         # used for view/edit office file via Office Online Server
         client_max_body_size 0;

         access_log      /var/log/nginx/seahub.access.log;
         error_log       /var/log/nginx/seahub.error.log;
    }
    location /seafhttp {
    rewrite ^/seafhttp(.*)$ $1 break;
    proxy_pass http://127.0.0.1:8082;
    client_max_body_size 0;
    proxy_set_header   X-Forwarded-For $proxy_add_x_forwarded_for;

    proxy_connect_timeout  36000s;
    proxy_read_timeout  36000s;
    proxy_send_timeout  36000s;

    send_timeout  36000s;

    access_log      /var/log/nginx/seafhttp.access.log;
    error_log       /var/log/nginx/seafhttp.error.log;
    }
    location /media {
        root /home/seafile/seafile-server-latest/seahub;
    }
}
```

Mit `sudo rm /etc/nginx/sites-enabled/default` und `sudo rm /etc/nginx/sites-available/default` löschen wir noch die Standardkonfiguration von `nginx`.

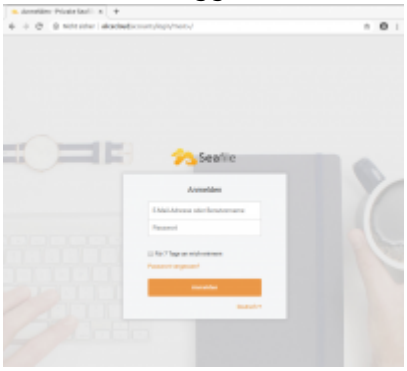
Mit `sudo nginx -t` können wir testen, ob die Konfiguration funktioniert und mit `sudo nginx -s reload` den Dienst `nginx` neustarten. Bei seafile müssen wir ebenso noch Anpassungen vornehmen:

1. Zum seafile-Benutzer mit `sudo su seafile` wechseln.
2. Mit `cd` ins Hauptverzeichnis und dann in den conf-Ordner wechseln mit `cd conf`.
3. Dort die Datei ccnet.conf anpassen `nano ccnet.conf` → Zeile hinzufügen: `SERVICE_URL = http://192.168.178.20` (IP anpassen).
4. Mit CTRL+S und CTRL+X speichern und schliessen.
5. Die seahub-Konfiguration öffnen mit `nano seahub_settings.py`.
6. Die Zeile `FILE_SERVER_ROOT = 'http://192.168.178.20/seafhttp'` anpassen und wieder mit CTRL+S und CTRL+X speichern und schliessen.
7. Die gunicorn-Konfiguration öffnen mit `nano gunicorn.conf.py`
8. Die Adresse `127.0.0.1:8000` zu `0.0.0.0` ändern.
9. Zum Hauptbenutzer wechseln mit `exit`.
10. seafile starten mit `sudo systemctl start seafile seahub`

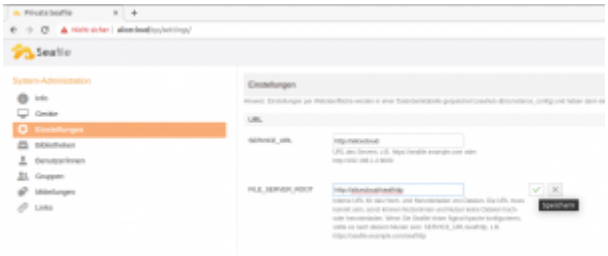
seafile im Netzwerk erreichen

Damit wir die Domain `alicecloud` verwenden können, müssen wir auf unserem PC noch die Hosts-Datei anpassen (für MacOS bzw. für Windows). Die Hosts-Datei öffnen und die Zeile: `192.168.178.20 alicecloud` hinzufügen (IP-Adresse anpassen). Seafile ist nun im Browser unter der Adresse `http://alicecloud` in eurem Netzwerk erreichbar.

Nun können wir uns mit den angegebenen Logindaten auf `http://alicecloud` einloggen.



Mit Klick auf den User-Avatar unter „System-Administration“ → „Einstellungen“ die SERVICE_URL auf `http://alicecloud` und FILE_SERVER_ROOT='`http://alicecloud/seafhttp`' anpassen. Wichtig: Die Änderung jeweils durch Klicken auf den grünen Haken bestätigen.

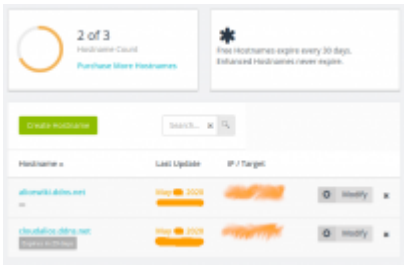


Fertig! Mun haben wir einen seafile-Server auf dem RPi installiert und können diesen nun im Browser im eigenen Netzwerk unter `http://alicecloud` erreichen.
Um seafile auch aus dem Internet zugänglich zu machen siehe Internetzugang.

Internetzugang

Bis jetzt ist der RPi nur übers eigene Netzwerk erreichbar, damit aber auch Schüler*innen oder andere Personen auf den Server zugreifen können, müssen wir noch den Zugang übers Internet einrichten.

Domains registrieren



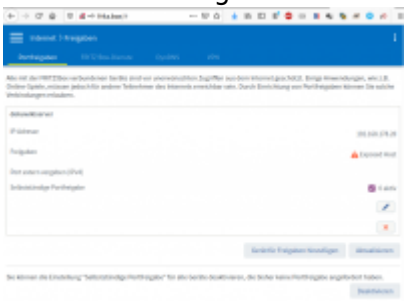
Dazu benötigen wir zuerst zwei Domains, eine für das Dokuwiki und eine für den seafile-Cloudspeicher. Es gibt verschiedene sogenannte DynDNS-Anbieter, die es ermöglichen eine Domain auf das eigene Netzwerk zuzuweisen. Ich habe no-IP gewählt. Da habe ich einen Account erstellt und zwei Domains registriert:

- `alicewiki.ddns.net` für das Dokuwiki und
- `cloudalice.ddns.net` für die seafile-Instanz.

Bei no-IP können kostenlos 3 Domains registriert werden. Diese haben jedoch eine begrenzte Laufzeit und müssen (falls nicht Premium erworben wird) jeweils einmal pro Monat im Account bei no-IP aktualisiert werden.

Portfreigabe am Router

Nun müssen wir noch den Port bzw. den Zugang zum RPi am Router freigeben. Dazu über `192.168.1.1` in die Routereinstellungen und unter „Portfreigaben“ den Port zum RPi freigeben (Port: 80 für die Installation, kann anschliessend wieder gelöscht werden und Port 443 dauerhaft).



Bei einer Fritz!Box ist die Einstellung in nebenstehenden Bild abgebildet. Hier kann nur generell eine Freigabe für eine IP-Adresse vorgenommen werden ohne spezifische Port-Nummer. Die Einstellung ist je nach Router unterschiedlich. Deshalb am Besten nach „Portfreigabe für Router xy “ googlen.

Portfreigabe nur vornehmen, wenn anschliessend auch die Verbindung gesichert wird.

Wenn die Portfreigabe aktiviert ist, sollten wir so schnell als möglich die Verbindung zu unserem Server sichern. Dies tun wir mit dem Certbot, welche eine sichere Verbindung mit `https` und SSL ermöglicht.

nginx für https konfigurieren (Dokuwiki)

Die Nginx-Konfiguration müssen wir für die https-Verbindung wie folgt anpassen:

1. Die Konfigurationsdatei umbenennen: `sudo mv /etc/nginx/sites-enabled/dokuwikizuhause.conf /etc/nginx/sites-enabled/alicewiki.ddns.net.conf` (Domainname anpassen).
2. Die Konfigurationsdatei öffnen: `sudo nano /etc/nginx/sites-enabled/alicewiki.ddns.net.conf`
3. Den Servername anpassen zu `alicewiki.ddns.net` (Domain anpassen).
4. Mit CTRL+S und CTRL+X die Änderungen speichern und die Datei schliessen.
5. Nun installieren wir Certbot, der uns ein Let's-Encrypt-Zertifikat für die sichere Verbindung mit dem Server erstellt:
    1. `sudo apt-get update`
    2. `sudo apt-get install python3-certbot-nginx`
6. nginx-Konfigurationsdatei anpassen:
    1. `sudo certbot --authenticator standalone --installer nginx -d alicewiki.ddns.net --pre-hook 'service nginx stop' --post-hook 'service nginx start'`
    2. Bei der Konfiguration eine gültige Email-Adresse angeben, an welche die Nachricht bei Ablauf des Zertifikats geschickt werden soll.
    3. Die Forderung Akzeptieren mit A und Enter.
    4. Keine Emails für Certbot-News mit N und Enter (ansonsten nach Wunsch mit Y).
    5. Eine automatische Weiterleitung auf die `https`-Verbindung mit dem Server einrichten: 2 und Enter.

Nun ist das Dokuwiki unter https://alicewiki.ddns.net übers Internet abrufbar. Falls Probleme mit der Konfiguration auftreten sollten: die Konfigurationsdatei `alicewiki.ddns.net.conf` sollte etwa so ausschauen:

```
server {
    server_name          alicewiki.ddns.net;
    root /var/www/dokuwiki;

    location / {
        index doku.php;
        try_files $uri $uri/ @dokuwiki;
    }
    location @dokuwiki {
        rewrite ^/_media/(.*) /lib/exe/fetch.php?media=$1 last;
        rewrite ^/_detail/(.*) /lib/exe/detail.php?media=$1 last;
        rewrite ^/_export/([^/]+)/(.*) /doku.php?do=export_$1&id=$2 last;
        rewrite ^/(.*) /doku.php?id=$1&$args last;
    }
    location ~ \.php$ {
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_pass unix:/run/php/php7.4-fpm.sock;
    }

    location ~ /(data|conf|bin|inc)/ {
        deny all;
    }

    listen [::]:443 ssl ipv6only=on; # managed by Certbot
```

```
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/alicewiki.ddns.net/fullchain.pem;
# managed by Certbot
    ssl_certificate_key
/etc/letsencrypt/live/alicewiki.ddns.net/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}
server {
    if ($host = alicewiki.ddns.net) {
        return 301 https://$host$request_uri;
    } # managed by Certbot


    listen               80;
    listen               [::]:80;
    server_name          alicewiki.ddns.net;
    return 404; # managed by Certbot


}
```

Mit `sudo nginx -t` kann geprüft werden, ob die Konfiguration von `nginx` funktioniert oder ob irgendwelche Fehler auftauchen.

nginx für https konfigurieren (seafile)

Für den seafile-Server müssen wir die nginx-Konfiguration für die https-Verbindung ebenfalls wie folgt anpassen:

1. Die Konfigurationsdatei umbenennen: `sudo mv /etc/nginx/sites-enabled/seafile.conf /etc/nginx/sites-enabled/cloudalice.ddns.net.conf` (Domainname anpassen).
2. Die Konfigurationsdatei öffnen: `sudo nano /etc/nginx/sites-enabled/cloudalice.ddns.net.conf`
3. Den Servername anpassen zu `cloudalice.ddns.net`.
4. Mit CTRL+S und CTRL+X die Änderungen speichern und die Datei schliessen.
5. Nun installieren wir Certbot, der uns ein Let's Encrypt Zertifikat für die sichere Verbindung mit dem Server erstellt.
    1. `sudo apt-get update`
    2. `sudo apt-get install certbot`
6. `nginx`-Konfigurationsdatei anpassen:
    1. `sudo certbot --authenticator standalone --installer nginx -d cloudalice.ddns.net --pre-hook 'service nginx stop' --post-hook 'service nginx start'`
    2. Bei der Konfiguration eine gültige Email-Adresse angeben, an welche die Nachricht bei Ablauf des Zertifikats geschickt werden soll.
    3. Die Forderung Akzeptieren mit A und Enter.
    4. Keine Emails für Certbot-News mit N und Enter (ansonsten nach Wunsch mit Y).

5. Eine automatische Weiterleitung auf die `https`-Verbindung mit dem Server einrichten: 2 und Enter.

7. Den seafile-Server unter https://cloudalice.ddns.net aufrufen und sich einloggen.
8. Unter „System-Administration“ → „Einstellungen“ die SERVER_URL und FILE_SERVER_ROOT anpassen mit https://cloudalice.ddns.net bzw. http://cloudalice.ddns.net/seafhttp.

Nun ist seafile unter https://cloudalice.ddns.net übers Internet abrufbar. Falls Probleme mit der Konfigurationsdatei auftreten sollten, die Konfigurationsdatei `cloudalice.ddns.net.conf` sollte etwa so aussehen:

```
server {
    server_name cloudalice.ddns.net;

    proxy_set_header X-Forwarded-For $remote_addr;

    location / {
         proxy_pass         http://127.0.0.1:8000;
         proxy_set_header   Host $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-Host $server_name;
         proxy_read_timeout  1200s;


         # used for view/edit office file via Office Online Server
         client_max_body_size 0;

         access_log      /var/log/nginx/seahub.access.log;
         error_log       /var/log/nginx/seahub.error.log;
    }
    location /seafhttp {
    rewrite ^/seafhttp(.*)$ $1 break;
    proxy_pass http://127.0.0.1:8082;
    client_max_body_size 0;
    proxy_set_header   X-Forwarded-For $proxy_add_x_forwarded_for;

    proxy_connect_timeout  36000s;
    proxy_read_timeout  36000s;
    proxy_send_timeout  36000s;

    send_timeout  36000s;

    access_log      /var/log/nginx/seafhttp.access.log;
    error_log       /var/log/nginx/seafhttp.error.log;
    }
    location /media {
        root /home/seafile/seafile-server-latest/seahub;
    }

    listen [::]:443 ssl; # managed by Certbot
```

```
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/cloudalice.ddns.net/fullchain.pem;
# managed by Certbot
    ssl_certificate_key
/etc/letsencrypt/live/cloudalice.ddns.net/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}
server {
    if ($host = cloudalice.ddns.net) {
        return 301 https://$host$request_uri;
    } # managed by Certbot


    listen 80;
    listen [::]:80;
    server_name cloudalice.ddns.net;
    return 404; # managed by Certbot


}
```

Sicherheit erhöhen

Da der RPi übers Internet erreichbar ist, sollten wir noch einige Sicherheitsmassnahmen ergreifen.

Passwort immer abfragen

Damit die Sicherheit noch etwas erhöht wird, werden wir einstellen, dass der RPi beim Befehl `sudo…` immer ein Passwort verlangt.

Dazu `sudo nano /etc/sudoers.d/010_pi-nopasswd` im Terminal eingeben und mit Enter öffnet sich ein Editor. Dort den Namen `pi` mit `alice` ersetzen und `NOPASSWD` zu `PASSWD` ändern. Mit `CTRL+S` die Änderungen speichern und den Editor mit `CTRL+X` schliessen.

Fail2Ban

Um verdächtige Verbindungen zu blockieren, gibt es ein nützliches Tool: `Fail2Ban`. Dieses Tool sucht in den angelegten Log-Dateien des RPi nach verdächtigen Zugriffen und blockiert diese. Es schützt den RPi somit z.B. vor Brute-force-Angriffen.

Zur Installation den Befehl `sudo apt-get install fail2ban -y` ausführen. Die Konfigurationsdatei mit dem Befehl `sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local` aktivieren und mit dem Editor öffnen: `sudo nano /etc/fail2ban/jail.local`.
Mit der Pfeiltaste oder `Page-Down` bis ans Ende der Datei scrollen und folgende Zeilen anfügen:

```
[seafile]
enabled  = true
port     = http, https
filter   = seafile-auth
logpath  = /home/seafile/logs/seahub.log
maxretry = 5
```

Mit `cd /etc/fail2ban/filter.d` in den Unterordner `filter.d` wechseln und eine Konfiguartionsdatei erstellen: `sudo nano seafile-auth.conf`. In die Datei folgende Zeilen einfügen, speichern und schliesen:

```
[INCLUDES]
before = common.conf

[Definition]
_daemon = seaf-server
failregex = Login attempt limit reached.*, ip: <HOST>
ignoreregex =
```

Nun Fail2Ban neu starten mit `sudo systemctl restart fail2ban` und mit `sudo fail2ban-client status` den Status prüfen. Die Ausgabe sollte etwa so aussehen:

```
Status
|- Number of jail:  2
`- Jail list:   seafile, sshd
```

Firewall einrichten

Es lässt sich eine einfache Firewall mit dem Befehl `sudo apt install ufw` installieren. Damit wir per `https` und `ssh` auf unseren Server zugreifen können, erlauben wir diese Zugriffe mit `sudo ufw allow ssh` und `sudo ufw allow 'Nginx Full'` . Mit `sudo ufw enable` aktivieren wir die Firewall und führen mit `sudo reboot` einen Neustart durch.

Mittels `sudo ufw status` kann der Status der Firewall abgefragt werden. Dieser sollte in etwa so aussehen:

```
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
Nginx Full                 ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
Nginx Full (v6)            ALLOW       Anywhere (v6)
```

Um noch eine Stufe mehr Sicherheit zu erlangen, kann zur Verbindung mit dem RPi per SSH die Passwortidentifikation deaktiviert und mit einer Key-Authentifizierung ersetzt werden (der RPi kann dann nur mit dem generierten Schlüssel über SSH erreicht werden). Siehe dazu den Abschnitt „Using key-based authentication“ unter Secure Raspberry Pi.

Bemerkungen

- Dieser Server ist beliebig erweiterbar. Evtl. liesse sich auch seafile noch in Dokuwiki integrieren…
- Ein Raspberry Pi 4 mit 4GB ist wahrscheinlich genügend performant – auch für mehrere Schulklassen.
- Um die SD-Karte etwas zu schonen und mehr Speicherplatz zu erhalten (mit der obigen Installation bleiben noch gut 26 GB freier Speicher), kann eine externe Festplatte als seafile-Datenspeicher dienen.

Quellen

Raspberry Pi OS
Remote-Access SSH
Secure Raspberry Pi
Dokuwiki
seafile-Manual
Dokuwiki install on Raspberry Pi
seafile Installation auf dem Raspberry Pi