

Nginx Reverse Proxy

Docker installieren

Nginx Proxymanager als Dockerimage erstellen

<https://nginxproxymanager.com/setup/#running-on-raspberry-pi-arm-devices>

Datei erstellen

docker-compose.yml Datei erstellen

MYSQL_PASSWORD anpassen

im Ordner der Docker Compose File folgenden Befehl absetzen:

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

Default Administrator User

Email: admin@example.com

Password: changeme

Immediately after logging in with this default user you will be asked to modify your details and change your password.


Upgrade zu neuer Version

```
docker-compose down
```

```
docker-compose pull
```

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

Proxy Stand 2023-06

 bit.hhml.selfhost.co Created: 20th March 2023	http://10.1.1.112:8080	Custom	Public	● Online	⋮
 books.hhml.selfhost.co Created: 20th March 2023	http://10.1.1.105:6875	Let's Encrypt	Public	● Online	⋮
 contacts.hhml.selfhost.co Created: 20th March 2023	https://10.1.1.10:25556	Let's Encrypt	Public	● Online	⋮
 dswoehr.hhml.selfhost.co Created: 20th March 2023	https://10.1.1.10:5001	Let's Encrypt	Public	● Online	⋮
 milec1oud.hhml.selfhost.co Created: 20th March 2023	http://10.1.1.75:8050	Let's Encrypt	Public	● Online	⋮
 nc.hhml.selfhost.co Created: 20th March 2023	http://10.1.1.124:80	Let's Encrypt	Public	● Online	⋮
 onlyoff.hhml.selfhost.co Created: 20th March 2023	https://10.1.1.2:8006	Let's Encrypt	Public	● Online	⋮
 paperless.hhml.selfhost.co Created: 20th March 2023	http://10.1.1.75:8050	Let's Encrypt	Public	● Online	⋮
 pho.hhml.selfhost.co Created: 31st March 2023	http://10.1.1.110:80	Let's Encrypt	Public	● Online	⋮
 projekt.hhml.selfhost.co Created: 20th March 2023	http://10.1.1.108:80	Let's Encrypt	Public	● Online	⋮
 umb.hhml.selfhost.co Created: 31st March 2023	http://10.1.1.110:8008	Let's Encrypt	Public	● Online	⋮
 webdav.hhml.selfhost.co Created: 20th March 2023	https://10.1.1.10:2006	Let's Encrypt	Public	● Online	⋮
 wiki.hhml.selfhost.co Created: 20th March 2023	http://10.1.1.41:80	Let's Encrypt	Public	● Online	⋮

Nginx manuelle Installation

Schritt 1: Installation von Nginx

```
sudo apt-get update  
sudo apt-get install nginx
```

Schritt 2: Konfiguration des Reverse Proxys

```
cd /etc/nginx/conf.d/
```

Erstellen der Konfigurationsdatei für den Reverse Proxy:

```
sudo nano reverse_proxy.conf
```

Folgenden Inhalt in die Datei

```
server {  
    listen 80;  
    server_name DOMAIN_NAME;  
  
    location / {  
        proxy_pass http://IP_ADDRESS:PORT;  
        proxy_set_header Host $host;  
        proxy_set_header X-Real-IP $remote_addr;  
    }  
}
```

Ersetzen Sie "DOMAIN_NAME" durch den Domainnamen oder die IP-Adresse des Servers, über den Sie auf die Computer im Netzwerk zugreifen möchten. Ersetzen Sie außerdem "IP_ADDRESS" durch die IP-Adresse des Computers, den Sie verfügbar machen möchten, und "PORT" durch den entsprechenden Port des Dienstes, auf den Sie zugreifen möchten.

Schritt 3: Neustart des Nginx-Servers

```
sudo service nginx restart
```

Schritt 4: Firewall-Einstellungen

Firewall-Einstellungen Wenn Sie eine Firewall auf Ihrem System verwenden (z. B. UFW), müssen Sie möglicherweise den Port öffnen, den Sie in der Konfigurationsdatei festgelegt haben. Verwenden Sie den folgenden Befehl, um den Port zu öffnen (ersetzen Sie "PORT" durch den tatsächlichen Port):

```
sudo ufw allow PORT
```

Schritt 5: Überprüfung des Reverse Proxys Geben Sie die Domain oder IP-Adresse des Servers, über den Sie auf die Computer im Netzwerk zugreifen möchten, in einen Webbrowser ein. Wenn alles korrekt konfiguriert ist, sollten Sie auf den Dienst zugreifen können, der auf dem Computer im Netzwerk läuft.

SSL-Zertifikat

Schritt 1: Installation des Certbot-Tools Certbot ist ein Open-Source-Tool, das die Einrichtung von SSL-Zertifikaten von Let's Encrypt automatisiert. Installieren Sie Certbot auf Ihrem Linux-System, indem Sie die Anweisungen auf der offiziellen Certbot-Website befolgen. Die genauen Schritte können je nach Linux-Distribution variieren.

Schritt 2: Konfiguration des Webserver. Stellen Sie sicher, dass Ihr Webserver (z.B. Nginx oder Apache) ordnungsgemäß für die Subdomain konfiguriert ist. Die Subdomain muss auf den richtigen Server verweisen und die erforderlichen Einstellungen für den SSL-Traffic zulassen.

Schritt 2a: Öffnen Sie das Terminal auf Ihrem Linux-System.

Schritt 2b: Führen Sie die folgenden Befehle aus, um Certbot zu installieren und das Nginx-Plugin zu aktivieren:

```
sudo apt update  
sudo apt install certbot python3-certbot-nginx
```

Diese Befehle aktualisieren zunächst die Paketlisten auf Ihrem System und installieren dann Certbot sowie das Nginx-Plugin für Certbot.

Schritt 2c: Überprüfen Sie, ob Certbot erfolgreich installiert wurde, indem Sie den Befehl `certbot --version` ausführen. Sie sollten eine Ausgabe sehen, die die installierte Version von Certbot anzeigt.

Schritt 2d: Konfiguration des Nginx-Plugins in Certbot: Certbot benötigt Informationen über die Nginx-Konfiguration, um das SSL-Zertifikat erfolgreich zu generieren. Führen Sie den folgenden Befehl aus, um das Nginx-Plugin in Certbot zu konfigurieren:

```
sudo certbot --nginx
```

Certbot wird die Nginx-Konfiguration analysieren und Ihnen dann eine Liste der verfügbaren Domains anzeigen, für die Sie ein SSL-Zertifikat erhalten können. Wählen Sie die gewünschte Subdomain aus, indem Sie die entsprechende Nummer eingeben und die Anweisungen befolgen.

Schritt 2e: Überprüfen Sie die Zertifikatserstellung: Certbot führt eine Herausforderung durch, um Ihre Kontrolle über die Subdomain zu überprüfen. Stellen Sie sicher, dass der Nginx-Server während dieses Prozesses erreichbar ist.

Certbot generiert das SSL-Zertifikat und speichert es auf Ihrem System. Die genauen Speicherorte variieren je nach Linux-Distribution und Nginx-Konfiguration. Certbot nimmt automatisch die erforderlichen Änderungen an Ihrer Nginx-Konfigurationsdatei vor, um das neu erstellte SSL-Zertifikat zu verwenden.

Schritt 2f: Neustart des Nginx-Servers: Starten Sie den Nginx-Server neu, damit die Konfigurationsänderungen wirksam werden:

```
sudo service nginx restart
```

Nach Abschluss dieser Schritte sollte der Nginx Reverse Proxy das gültige SSL-Zertifikat verwenden und den Datenverkehr über HTTPS verschlüsseln. Überprüfen Sie dies, indem Sie die Subdomain in einem Webbrowser öffnen und sicherstellen, dass das Zertifikat korrekt funktioniert.

Schritt 3: Zertifikatserstellung mit Certbot Führen Sie den folgenden Befehl aus, um Certbot zu verwenden und ein SSL-Zertifikat für den Reverse Proxy zu erhalten:

```
sudo certbot certonly --nginx -d subdomain.example.com
```

Ersetzen Sie "subdomain.example.com" durch Ihre eigene Subdomain.

Certbot interagiert mit Nginx, erkennt die Konfiguration der Subdomain und fordert Sie auf, den gewünschten Domainnamen und die E-Mail-Adresse einzugeben. Befolgen Sie die Anweisungen, um den Prozess abzuschließen.

Schritt 4: Konfiguration des Nginx Reverse Proxys Certbot sollte automatisch die erforderlichen Änderungen an Ihrer Nginx-Konfigurationsdatei vornehmen, um das neu erstellte SSL-Zertifikat zu verwenden. In der Regel werden die Änderungen in einer separaten Konfigurationsdatei unter `/etc/nginx/conf.d/` vorgenommen.

Überprüfen Sie die Nginx-Konfigurationsdateien und stellen Sie sicher, dass der Reverse Proxy ordnungsgemäß auf das SSL-Zertifikat verweist. Normalerweise werden die SSL-Einstellungen in einer `server`-Block-Konfiguration wie folgt aussehen:

```
server {  
    listen 443 ssl;  
    server_name subdomain.example.com;  
  
    ssl_certificate /etc/letsencrypt/live/subdomain.example.com/fullchain.pem;  
    ssl_certificate_key /etc/letsencrypt/live/subdomain.example.com/privkey.pem;  
  
    location / {  
        proxy_pass http://IP_ADDRESS:PORT;  
        proxy_set_header Host $host;  
        proxy_set_header X-Real-IP $remote_addr;  
    }  
}
```

Stellen Sie sicher, dass die Pfade zu den Zertifikatsdateien (`ssl_certificate` und `ssl_certificate_key`) korrekt angegeben sind und auf die von Certbot generierten Zertifikate verweisen.

Schritt 5: Neustart des Nginx-Servers Starten Sie den Nginx-Server neu, damit die Konfigurationsänderungen wirksam werden:

```
sudo service nginx restart
```

Nach Abschluss dieser Schritte sollte der Nginx Reverse Proxy das gültige SSL-Zertifikat verwenden und den Datenverkehr über HTTPS verschlüsseln. Überprüfen Sie dies, indem Sie die Subdomain in einem Webbrowser öffnen und sicherstellen, dass das Zertifikat korrekt funktioniert.

Konfigurationsdatei für mehrere Subdomains

Wie sieht die Config-Datei aus, wenn mehrere Rechner durchgeleitet werden sollen:

```
http {
    # HTTPS-Server für Port 443
    server {
        listen 443 ssl;
        server_name subdomain1.example.com;

        ssl_certificate /etc/letsencrypt/live/subdomain1.example.com/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live/subdomain1.example.com/privkey.pem;

        location / {
            proxy_pass http://IP_ADDRESS1:PORT1;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
        }
    }

    # HTTPS-Server für Port 443, zweite Subdomain
    server {
        listen 443 ssl;
        server_name subdomain2.example.com;

        ssl_certificate /etc/letsencrypt/live/subdomain2.example.com/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live/subdomain2.example.com/privkey.pem;

        location / {
            proxy_pass http://IP_ADDRESS2:PORT2;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
        }
    }
}
```

```
}

# Weitere server-Blöcke für zusätzliche Subdomains und Reverse-Proxy-Verbindungen
# ...

# Weitere Nginx-Konfigurationseinstellungen
# ...

}
```

Revision #7

Created 20 March 2023 21:32:38 by Hermann

Updated 6 October 2024 10:48:31 by Hermann