

Caddyfile

- Grundlegende Reverseproxyfunktion
- Andere Caddy-Konfigurationen importieren
- Vorhandenes Zertifikat in Caddy verwenden
- Log-Datei für virtuellen Host in Caddy schreiben
- IP-Bereiche für Zugriffe auf Caddy Server sperren
- Automatische HTTP auf HTTPS Umleitung einrichten

Grundlegende Reverseproxyfunktion

Struktur ohne TLS

```
<FQDN> {  
  reverse_proxy <ip-oder-hostname>:<port>  
}
```

Struktur mit internen TLS Zertifikat

```
<FQDN> {  
  reverse_proxy <ip-oder-hostname>:<port>  
  tls internal  
}
```

Andere Caddy-Konfigurationen importieren

Wir können, um die Konfiguration besser zu organisieren, einzelne Konfigurationsdateien für unsere virtuellen Hosts erstellen. Diese importieren wir dann mit einem Befehl in unsere **Caddyfile**.

Wir fügen einfach am Ende unserer **Caddyfile** den folgenden Befehl ein:

```
import /etc/caddy/conf/*.caddy
```

Es werden dann alle Dateien mit der `.caddy` Endung aus dem **conf-Verzeichnis** in die **Caddyfile** geladen. Das ermöglicht dann unsere Dateien besser zu organisieren.

Vorhandenes Zertifikat in Caddy verwenden

Wir können für unseren **Caddy-Webserver** auch vorhandene **TLS-Zertifikate** verwenden. Dazu erstellen wir einen **virtuellen Host** und fügen dem die `tls` Option hinzu. Hinter der Option geben wir dann noch an, wo Caddy das **Zertifikat** und den **Zertifikatsschlüssel** findet.

```
www.website.de {  
    reverse_proxy 192.168.10.2:80  
  
    tls /opt/certificates/website.de/cert.pem /opt/certificates/website.de/key.pem  
}
```

Log-Datei für virtuellen Host in Caddy schreiben

Man kann in der Konfiguration des virtuellen Hosts im Caddy hinterlegen, ob eine **Log-Datei** geschrieben werden soll. Um diese dann mit dem **Prometheus Log Exporter** auszuwerten, soll die Log-Datei als **JSON** formatiert werden.

```
www.website.de {  
    reverse_proxy 192.168.10.2  
  
    tls /opt/certificates/website.de/cert.pem /opt/certificates/website.de/key.pem  
  
    log {  
        output file /var/log/caddy/website.access.log  
        format json  
    }  
}
```

IP-Bereiche für Zugriffe auf Caddy Server sperren

Um IP-Bereiche für den Zugriff auf einen virtuellen Host im **Caddy Webserver** zu blockieren, müssen wir in die Konfiguration der **Caddyfile** den Befehl `@blocked not remote ip <ip-subnet1> <ip-subnet2> <...>` verwenden.

Im Anschluss müssen wir nur den "Verweigerungstext" angeben. Schlussendlich kann die Konfiguration wie folgt aussehen:

```
www.website.de {  
    reverse_proxy 192.168.10.2:80  
    @blocked not remote_ip 192.168.20.0/24  
    respond @blocked "Zugriff verweigert!" 403  
  
    tls /opt/certificates/website.de/cert.pem /opt/certificates/website.de/key.pem  
  
    log {  
        output file /var/log/caddy/website.access.log  
        format json  
    }  
}
```

Automatische HTTP auf HTTPS Umleitung einrichten

Einleitung

In diesem kurzen Artikel geht es darum, wie wir in unserer Caddyfile einrichten können, dass jeglicher HTTP-Verkehr automatisch auf HTTPS umgeleitet wird. Dazu müssen wir unsere Konfigurationsdatei nur minimal anpassen.

Durchführung

Im ersten Schritt öffnen wir unsere Caddyfile mit einem Editor unserer Wahl. Ich verwende immer `nano` dazu.

```
nano /etc/caddy/Caddyfile
```

Dort fügen wir folgenden Inhalt ein:

```
http:// {  
    redir https://{host}{uri}  
}
```

Sobald wir unsere Datei angepasst haben, lassen wir einmal Caddy überprüfen, ob unsere Konfiguration verwendbar für Caddy ist. Dazu führen wir den folgenden Befehl aus:

```
caddy validate --config /etc/caddy/Caddyfile
```

Info: Wenn als Ausgabe "Valid Configuration" erscheint, haben wir alles richtig gemacht.

Zum Schluss aktivieren wir noch unsere angepasste Konfiguration:

```
systemctl reload caddy
```