Training Apache2

Agenda

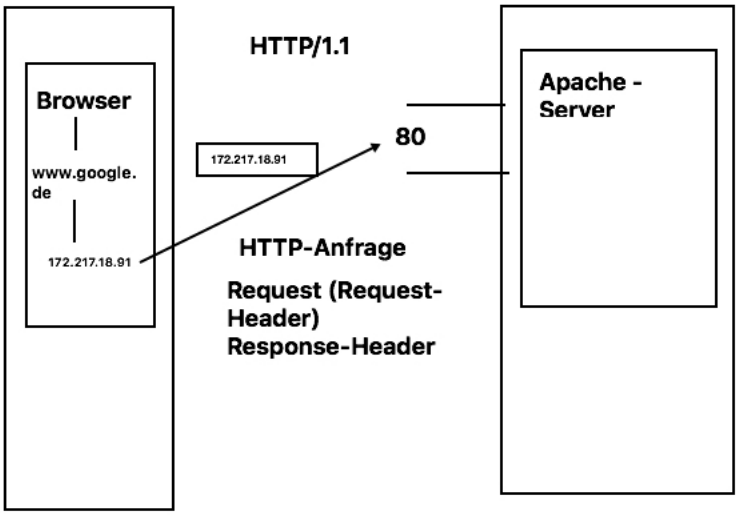
1. Einführung
    - Apache http-requests
2. Installation
    - Installation unter RHEL/RockyLinux
    - Starten/Stoppen/Aktivieren und weitere hilfreiche systemctl - Befehle
    - Lauscht mein Server nach draussen
    - Die Firewall zähmen/freischalten
3. Administration
    - Konfigurationsdatei mit httpd -t prüfen
    - Alle geladenen Module finden
    - Mehrere Instanzen mit systemd starten
    - Rechte debuggen mit der Bash für apache
    - Default-VirtualHost richtig konfigurieren - Achtung !
    - Alias für Fehlerseite in VirtualHost
4. Monitoring
    - Status aktivieren
5. Rewriting / Resetting
    - Rewrite Header on if
    - URL's manpulieren/umschreiben mit mod_rewrite
6. Performance / Optimierung
    - Optimierung mpm event
7. Authentication
    - Simple Textbased Authentication
    - Authentication with MySQL/MariadDB
8. SSL
    - SSL mit letsencrypt
    - Client Base certificate
9. SELinux
    - Neuen unbekannten Port freischalten, z.B. 86
    - Neues DocumentRoot verwenden
10. Absichern
    - Standard-Apache-Seite deaktivieren (RHEL/Rocky)
    - Standard-Fehlerseiten konfigurieren
    - Verzeichnislisting-autoindex deaktivieren/härten
    - Keine Apache Version kommunizieren
    - Rechte härten
11. Logs
    - Journal persistent setzen
    - Piped Logging
12. Proxy
    - Reverse Proxy
    - Reverse Proxy Balancer
13. Tipps & Tricks
    - In den Root-Benutzer wechseln
    - Wo bin ich ?
    - Nach Direktive suchen
    - Praktische Ausgabe von langen Seiten - less
    - cloudinit-script zum Anlegen von Server auf digitalocean
14. Prozesse
    - Prozesse anzeigen - ps/pstree -p
    - Prioritäten und NiceNess
15. Logs/Loganalyse
    - Logfile beobachten
    - Dienste debuggen
    - Journal analysieren
16. Dienste/Runlevel(Targets verwalten)

- Die wichtigsten systemctl/service

17. Hilfe
    - Hilfe zu Befehlen
18. Firewall
    - firewalld
    - Scannen und Überprüfen mit telnet/nmap
19. Netzwerk/Dienste
    - Hostname setzen
20. Dokumentation
    - Apache module mit Direktiven
    - Apache Konfigurationen (Directives) für alle Module
    - Linux Security - PDF

Einführung

Apache http-requests



Installation

Installation unter RHEL/RockyLinux

```
##
dnf install httpd
```

Starten/Stoppen/Aktivieren und weitere hilfreiche systemctl - Befehle

Apache starten / stoppen / aktivieren

```
## Apache wird nach der Installation nicht standardmäßig gestartet in Centos/RHEL
systemctl status httpd
systemctl start httpd
systemctl status httpd
systemctl is-enabled httpd
## Rückgabewert des letzten Befehls
## Wenn nicht aktiviert kommt eine 1 raus
echo $?

systemctl enable httpd
systemctl is-enabled httpd
## Jetzt kommt eine 0 raus
echo $?
```

systemctl Beispiele

```
## Wie heisst der Dienst / welche Dienste gibt es ? (nur wenn der service aktiviert ist).
systemctl list-units -t service
## für apache
systemctl list-units -t service | grep ^httpd
## die Abkürzung
systemctl -t service | grep ^httpd

## Wie finde ich einen service, der noch nicht aktiviert ist ?
systemctl list-unit-files -t service | grep httpd

## Rebooten des Servers
## verweist auf systemctl
reboot
systemctl reboot
shutdown -r now
```

```
## Halt (ohne Strom ausschalten)
halt
systemctl halt
shutdown -h now

## Poweroff
poweroff
systemctl poweroff
```

Wie sehe ich, wie ein Service konfiguriert ist / Dienstekonfiguration anzeigen ?

```
## z.B. für Apache2
systemctl cat apache2.service
```

Wie kann ich rausfinden, wie die runlevel als targets heissen ?

```
cd /lib/systemd/system
root@ubuntu2004-104:/lib/systemd/system# ls -la run*target
lrwxrwxrwx 1 root root 15 Jan  6 20:47 runlevel0.target -> poweroff.target
lrwxrwxrwx 1 root root 13 Jan  6 20:47 runlevel1.target -> rescue.target
lrwxrwxrwx 1 root root 17 Jan  6 20:47 runlevel2.target -> multi-user.target
lrwxrwxrwx 1 root root 17 Jan  6 20:47 runlevel3.target -> multi-user.target
lrwxrwxrwx 1 root root 17 Jan  6 20:47 runlevel4.target -> multi-user.target
lrwxrwxrwx 1 root root 16 Jan  6 20:47 runlevel5.target -> graphical.target
lrwxrwxrwx 1 root root 13 Jan  6 20:47 runlevel6.target -> reboot.target
```

Welche Dienste sind aktiviert/deaktiviert

```
systemctl list-unit-files -t service
```

Dienste bearbeiten

```
systemctl edit sshd.service
## Dann eintragen
[Unit]
Description=Jochen's ssh-server
## Dann speichern und schliessen (Editor)

systemctl daemon-reload
systemctl status
```

Targets (wechseln und default)

```
## Default runlevel/target auslesen
systemctl get-default
## in target wechseln
systemctl isolate multi-user
## Default target setzen (nach start/reboot)
systemctl set-default multi-user
```

Alle Target anzeigen in die ich reinwechseln kann (isolate)

```
## Ubuntu
grep -r "AllowIsolate" /lib/systemd/system
/lib/systemd/system/reboot.target
...
...
...
systemctl isolate reboot.target
```

Dienste maskieren, so dass sie nicht gestartet werden können

```
systemctl mask apache2
## kann jetzt gestartet werden
systemctl start apache2

## de-maskieren
systemctl unmask apache2
## kann wieder gestaret werden
systemctl start apache2
```

systemctl Cheatsheet

- https://access.redhat.com/sites/default/files/attachments/12052018_systemd_6.pdf

Lauscht mein Server nach draussen

```
## Zeigt alle ports an auf die gelauscht wird (ipv4)
## Ist httpd in der Liste
lsof -i
## alternative
netstat -tupel

## Wie heisst dort Port als Nummer ?
cat /etc/services | grep "^httpd "
```

Die Firewall zähmen/freischalten

Short: Aktivieren

```
firewall-cmd --add-service=http
firewall-cmd --runtime-to-permanent

## oder
firewall-cmd --add-service=http --permanent
firewall-cmd --reload
```

Long version

```
systemctl status firewalld
man firewall-cmd
firewall-cmd --state
firewall-cmd --get-zones
firewall-cmd --get-active-zones
## alle einstellungen der public zone
firewall-cmd --list-all

## Welche Services gibt es ?
firewall-cmd --get-services

## Services  des Distributors
ls -la /usr/lib/firewalld/services
cat /usr/lib/firewalld/services/http.xml

## Aktuelle abgespeicherte Konfiguration (permanent)
cat /etc/firewalld/zones/public.html

## Welches Backend wird verwendet
cat /etc/firewalld/firewalld.conf | grep -i Backend
```

Administration

Konfigurationsdatei mit httpd -t prüfen

- httpd-t.md

Alle geladenen Module finden

```
for i in /etc/httpd/conf.modules.d/*; do cat $i | grep LoadModule | sed 's/^ *//' | cut -d' ' -f2 ; done

## or simply
httpd -M | cut -d' ' -f2
```

```
## Finds all IfModule entries
for i in $(httpd -M | cut -d' ' -f2); do  grep -r $i /etc/httpd | grep -v LoadModule  ; done
```

Mehrere Instanzen mit systemd starten

Walkthrough

```
#### Schritt 1: Original conf kopiert
cd /etc/httpd/conf
cp -a httpd.conf 1.conf


#### Schritt 2: Konfig anpassen

## Zu Testzwecken nur wichtigste Konfigurationen geändert.
## In Produktion auch DocumentRoot / Logs und anderes IncludeOptional conf.d/* - Verzeichnis
## conf.modules.d koennen sie gemeinsam haben
```

```
## Zeile mit Listen geaendert
## von
## Listen 80
## in
Listen 81

## Eigenes Run Verzeichnis festlegen und eigenes PID - File
## Load config files in the "/etc/httpd/conf.d" directory, if any.
## IncludeOptional conf.d/*.conf
DefaultRuntimeDir /run/httpd/instance-${HTTPD_INSTANCE}
PidFile /run/httpd/instance-${HTTPD_INSTANCE}.pid

## conf.d auskommentieren oder anderen Pfad (neues Verzeichnis wählen)

#### Schritt 3: Dienst starten und evtl. aktiveren
systemctl start httpd@1.service
systemctl enable httpd@1.service
```

Herleitung

```
##
## where to find the documentation
systemctl status httpd.service | grep Doc
man httpd.service
## Ausschnitt
 To allow multiple instances of httpd to run simultaneously, a number of
       configuration directives must be changed, such as PidFile and
       DefaultRuntimeDir to pick non-conflicting paths, and Listen to choose
       different ports. The example configuration file
       /usr/share/doc/httpd/instance.conf demonstrates how to make such
       changes using HTTPD_INSTANCE variable.


## Schritt 2. instance.conf schauen
less /usr/share/doc/httpd/instance.conf
## und rauskopieren
DefaultRuntimeDir /run/httpd/instance-${HTTPD_INSTANCE}
PidFile /run/httpd/instance-${HTTPD_INSTANCE}.pid

## Schritt 3. /etc/httpd/conf.d/1.conf aendern
## ans Ende anfuegen
DefaultRuntimeDir /run/httpd/instance-${HTTPD_INSTANCE}
PidFile /run/httpd/instance-${HTTPD_INSTANCE}.pid
```

Rechte debuggen mit der Bash für apache

```
[root@server1 www]# sudo su - apache -s /bin/bash

## Jetzt können wir durch die Verzeichnis wandern (mit cd)
und gucken, ob wir an die gewünschte Stelle kommen (ohne permission denied)

cd /var/www
cd html
-bash: cd: html: Keine Berechtigung
```

Default-VirtualHost richtig konfigurieren - Achtung !

Wichtig !!

```
Die VirtualHost - config für den Default - Host muss immer als
erstes von Apache geparsed werden, weil der Eintrag den Default darstellt.

Einfach realisierbar, in dem man diesen 0-default.conf nennt.

 cd /etc/httpd/sites-enabled/
[root@server1 sites-enabled]# pwd
/etc/httpd/sites-enabled
[root@server1 sites-enabled]# ls -la
insgesamt 16
drwxr-xr-x. 2 root root  95 26. Jan 13:39 .
drwxr-xr-x. 6 root root 126 26. Jan 10:11 ..
-rw-r--r--. 1 root root 177 26. Jan 10:32 00-default.conf
-rw-r--r--. 1 root root 254 26. Jan 11:46 bw.de.conf
-rw-r--r--. 1 root root 261 26. Jan 13:39 casino.bw.de.conf
```

```
-rw-r--r--. 1 root root 251 26. Jan 13:36 jobs.bw.de.conf

Default Eintrag ohne Domain, dann greift dies als Default
(Achtung: unbedingt anlegen, Server-Config greift hier nicht !!! )

## vi /etc/httpd/sites-enabled/0-default.conf
<VirtualHost *:80>
   DocumentRoot /var/www/html
   ErrorLog ${APACHE_LOG_DIR}/default-error.log
   CustomLog ${APACHE_LOG_DIR}/default-access.log combined
</VirtualHost>
```

```
## Modifizierte httpd.service verwendet
## damit APACHE_LOG_DIR funktioniert (standardmäßig)
## systemctl edit httpd.service
## schreibt override datei
[Service]
Environment=APACHE_LOG_DIR=/var/log/httpd

systemctl restart httpd.service
```

```
### Alias für Fehlerseite in VirtualHost
```

vi /etc/httpd/sites-enabled/jobs.bw.de.conf

<VirtualHost *:80> ServerName jobs.bw.de ServerAlias www.jobs.bw.de

DocumentRoot /var/www/jobs.bw.de/html ErrorLog ${APACHE_LOG_DIR}/jobs.bw.de-error.log CustomLog ${APACHE_LOG_DIR}/jobs.bw.de-access.log combined ErrorDocument 404 /fehler Alias /fehler /var/www/html/404.html

oder direkte weiterleitung auf fehlerseite

wird ein browser zurückgegeben und browser macht redirect

server muss domain nicht auflösen können, erst der browser

ErrorDocument 404 http://www.casino2.bw.de/404.html

mkdir -p /var/www/html echo "

Personal 404

" > /var/www/html/404.html

systemctl restart httpd

```
## Monitoring

### Status aktivieren


### Aktivieren
```

/etc/httpd/sites-enabled/00-default.conf

<Location "/server-status"> SetHandler server-status Require ip 192.168.56.1

systemctl restart httpd

```
### Extended Status
```

Enabled by default in RHEL/Rocky

Refer to mod_status

ExtendedStatus - "On" to track extended status information, "Off" to disable

For Performance reasons disable it if not needed

on possible in ServerConfiguration

ExtendedStatus off

```
### Ref:

  * https://httpd.apache.org/docs/2.4/mod/mod_status.html

## Rewriting / Resetting

### Rewrite Header on if


### Example
```

Set response header if url contains 'test'

<If "%{REQUEST_URI} =~ /test/"> Header set MegaHeader 'Cool stuff'

```
### Great Examples

  * There are some great examples for Expression here:
    * [Examples Regex](https://httpd.apache.org/docs/2.4/expr.html#examples)

### Refs:


  * [Expression - Important reference](https://httpd.apache.org/docs/2.4/expr.html)


### URL's manpulieren/umschreiben mit mod_rewrite



### Rewriting aktivieren
```

RewriteEngine on

```
### Rewrite - bessere Alternative ?
```

Rewriting sollte das last resort sein,

Wenn man eine Alternative hat, sollte man dieser verwendet,

weil rewriting weniger performant

z.B. bei Alias / Redirect

```
### Mod_rewrite schwierig zu debuggen.
```

Alternativ kann sein, alle Anfragen immer an das gleiche Script zu leitern und dort zu verarbeiten, z.B. mit redirect

Warum ? Im Script kann man leichter debuggen.

```
### Logging einschalten
```

highest loglevel -

LogLevel rewrite:trace8

```
### Beispiel 1: alte Domain auf Neue Domain umleiten
```

Redirect all pages from olddomain.com

to newdomain.com

RewriteEngine on RewriteCond %{HTTP_HOST} ^jobs.jochen.t3isp.de$ [OR] RewriteCond %{HTTP_HOST} ^www.jobs.jochen.t3isp.de$

$1 bezieht sich auf die 1. Klammer

http://www.jobs.jochen.t3isp.de/test.html

^(.*)$ = /test.html

RewriteRule ^(.*)$ http://casino.jobs.jochen.de/$1 [R=301,L]

```
### Beispiel 2: Unterordner erzwingen
```

RewriteEngine On RewriteRule ^$ /folder/ [R=301,L]

```
### Beispiel 3: Unterseite erzwingen
```

RewriteEngine On RewriteRule ^$ /unterseite.html [R=301,L]

```
### Beispiel 4: Bestimmte Dateien nicht erlauben
```

##Do not allow these file types to be called RewriteEngine on RewriteRule .*.(jpg|jpeg|gif|png|bmp|exe|swf)$ - [F,NC]

```
### Beispiel 5: Rewrite only, if it does not exist
```

RewriteEngine On

RewriteCond %{REQUEST_FILENAME} !-f RewriteCond %{REQUEST_FILENAME} !-d RewriteCond %{REQUEST_FILENAME} !-l

RewriteRule .* index.php [L]

```
### Ref:

  * https://httpd.apache.org/docs/trunk/rewrite/intro.html

## Performance / Optimierung

### Optimierung mpm event


### Ziel

  * mehr Anfragen pro Sekunden(Requests/Sec)
  * Stabiles System

### Grundberechnung (Variante 1)

   * Aapche und php-fpm
```

Estimating Thrashing (Prügel) - Point (hier fängt er an auszulagern)

free -h (total memory - buffer/cache - Reserved) / Average Apache

Wie ist es mit php: 5 Kind-Prozesse,pro Prozess (pool): 18,5 MB

Beispielrechnung:

Schritt1: Speicher für Apache ermitteln

800MB Arbeitsspeicher

- # 308MB buffer/cache Betriebssystem

  492MB Speicher für alle Dienste zu vergeben

- # 100MB für 5 Prozesse php-fpm

- 400MB für Apache

Schritt2: Wieviel Kind-Prozesse sind möglich

1. Miniscrhitt: Ermittlung der durchschnittlichen Größe eines Apache - Prozesses: rund 16,5 MB = grosszügig 20 MB
2. Minischritt: Wieviel Kind/Child-Prozesse möglich 400MB / 20 MB = Anzahl MÖGLICHE Kindprozesse = ca 20
3. Aus Spaß Threads: 25 Threads pro Prozess: 500 Threads

```
### Ausgewogenheitsberechnung
```

pro Kind Prozess von Apache (Standard) 25 Threads d.h. aktuell 100 Threads

demgegenüber 5 Prozesse php d.h. hoch ziehen auf 100 d.h. 100 * 18 MB = 1,8 GB

d.h. unter 3 GB Arbeitsspeicher (1GB vorhanden aktuell) brauchen wir hier garnicht anfangen

```
### Grundberechnung (Variante 2)

  * Apache und vielleicht noch einen kleinen Dienst (undefiniert)
```

Beispielrechnung:

Schritt1: Speicher für Apache ermitteln

2000MB Arbeitsspeicher

- **308MB buffer/cache Betriebssystem**

  1892MB Speicher für alle Dienste zu vergeben

- **500MB 25% Pauschal für andere Prozesse**

- 1392MB für Apache

Schritt2: Wieviel Kind-Prozesse sind möglich

1. Miniscrhitt: Ermittlung der durchschnittlichen Größe eines Apache - Prozesses: rund 16,5 MB = grosszügig 20 MB
2. Minischritt: Wieviel Kind/Child-Prozesse möglich 1300MB / 20 MB = Anzahl MÖGLICHE Kindprozesse = 60 Apache prozesse
3. Aus Spaß Threads: 25 Threads pro Prozess: 1500 Threads

```
### Sind soviele Threads hilfreich ?
```

Was sind begrenzende Faktoren (bottleneck) (nur Apache) ?

o i/o = schreiben und lesen auf und von Platte o CPU (wie schnell CPU und wiele Operation können gleichzeitig verarbeiten)

Was sind Testschritte.

1. System unter Last setzen (ab - apache benchmark -c (gleichzetigie Prozesse -n wieviele Abfragen)

Wie sehe ich, ob das Sinne soviele Threads zu haben.

top

```
### CPU gebundene Last erkennen und handeln
```

Habe ich einen io-gebunden oder cpu-gebundene Last

CPU-gebundene Last

Was heisst das ?

Die CPU pfeift das aus dem letzten Loch Auch, wenn ich eine schnellere Platte habe, nützt das nix, weil die zu langsame CPU der begrenzende Faktor ist

Wie finde ich das heraus ?

top

1. In der Übersicht: CPus: us und u.U. sy wert Hoch
2. load average hoch: Spalte 1: Wieviele Prozesse warteten im Durchrschnitt auf Ausführung durch die CPU in der letzten Minute Spalte 2: Das gleiche wie 1 nur in den letzten 5 Minuten Spalte 3: Das gleiche wie 1 nur in den letzten 15 Minuten
3. In CPU-Zeile: wa-wert ist 0 oder niedrig.

Was tun ?

A) Schnellere CPU kaufen (horizontal skalieren) oder B) Weitere Systeme zu Entlasten einbinden (z.B. Proxy mit Backend) C) Varnish-Cash -> System durch vorgeschaltet Systeme entlassen

```
### IO gebundene Last erkennen und handeln
```

Habe ich einen io-gebunden oder cpu-gebundene Last

IO-gebundene Last

Was heisst das ?

Die CPU wartet auf die Festplatte, die Festplatte knirscht und kann nicht mehr abarbeiten, sie ist durch die Arbeit die Arbeit voll ausgelastet.

Wie finde ich das heraus ?

top

1. In der Übersicht: CPus: us und u.U. sy wert Hoch
2. load average hoch: Spalte 1: Wieviele Prozesse warteten im Durchrschnitt auf Ausführung durch die CPU in der letzten Minute Spalte 2: Das gleiche wie 1 nur in den letzten 5 Minuten Spalte 3: Das gleiche wie 1 nur in den letzten 15 Minuten

-> 3) In CPU-Zeile: wa-wert ist HOCH oder niedrig. <-

Was tun ?

A) Schnellere Platte kaufen (hdd -> ssd - platte) B) Weitere Systeme zu Entlasten einbinden (z.B. Proxy mit Backend) C) Varnish-Cash -> System durch vorgeschaltet Systeme entlassen

```
### Werte einstellungen
```

Welche Kind am Starten werden restarten/gestartet wird

Default 3

StartServers

Hard limit für Thread pro Kind-Prozess

Wieviele ThreadPerChild (Threads pro Kind kann ich maximal konfigurieren)

Standardmäßig: 64

ThreadLimits

Wieviel Threads per Child (max. ThreadLimits) werden beim Bereitstellen

eines neune Child erstellt

Standard ist 25

25 ist good to go.

Wenn höher unbedingt

ThreadsPerChild

Maximale Zahl an Kind-Prozessen

z.B. 4 Prozesse (die Threads sind damit nicht gemeint)

ServerLimit

Maximale Anzahl aller Threads über alle jobs

MaxRequestWorkers = 200 # vorher MaxClient vor 2.3 aktuell geht noch beides

MinSpareThreads

unbeschäftigte Threads, werden hochgefahren, wenn nicht genügen unbeschäftigte

Threads vorhanden sind.

MaxSpareThreads

```
## Authentication

### Simple Textbased Authentication


### Example config
```

Schritt 1: Passwort - Datei erstellen

Ubuntu / Debian

htpasswd -c /etc/apache2/.htpasswd kurs htpasswd /etc/apache2/.htpasswd another_user

RHEL / Rocky - password

htpasswd -c /etc/httpd/.htpasswd kurs htpasswd /etc/httpd/.htpasswd another_user

Schritt 2:Virtual Host - Konfiguration

<VirtualHost *:80> ServerAdmin webmaster@localhost DocumentRoot /var/www/html ErrorLog /var/log/httpd/error.log CustomLog /var/log/httpd/access.log combined

```
<Directory "/var/www/bw.de/html/private/">
    AuthType Basic
    AuthName "Bitte Einloggen:"
    AuthUserFile /etc/httpd/.htpasswd
    # AuthUserFile /etc/apache2/.htpasswd
    Require valid-user
</Directory>
```

Schritt 3:

systemctl restart http

Schritt 4: Testen - Ein Login-Prompt muss kommen

Browser: https://jochen.t3isp.de/private/

```
### Ref:

  * https://www.digitalocean.com/community/tutorials/how-to-set-up-password-authentication-with-apache-on-ubuntu-14-04

### Authentication with MySQL/MariadDB


### Ref:

  * https://www.howtoforge.de/anleitung/wie-man-verzeichnisse-mit-mod-authn-dbd-und-mysql-auf-dem-apache-mit-einem-passwort-schtzt-debian-8/

## SSL

### SSL mit letsencrypt


### Step 0:
```

dnf install -y httpd systemctl enable httpd systemctl start httpd

```
### Step 1: setup virtual host
```

/etc/httpd/conf/httpd.conf

IncludeOptional sites-enabled/*.conf

x = 1 to 7

apx.t3isp.de

mx.t3isp.de

mkdir /etc/httpd/sites-enabled

cd /etc/httpd/sites-enabled

vi ap1.t3isp.de.conf

<VirtualHost *:80> ServerName www.ap1.t3isp.de ServerAlias ap1.t3isp.de DocumentRoot /var/www/ap1.t3isp.de/html ErrorLog /var/log/httpd/ap1-t3isp-de-error.log CustomLog /var/log/httpd/ap1-t3isp-de-access.log combined

/var/www mkdir -p ap1.t3isp.de/html cd ap1.t3isp.de/html/ echo "ich bin ap1 von jochen" > index.html

```
### Install certbot (Centos 8)
```

dnf install -y epel-release && dnf install -y certbot python3-certbot-apache mod_ssl

```
### Variant 1: Attention: virtual host - domain must be different than hostname
```

e.g. ap1.t3isp.de (virtual host domain) != hostname

if this is not the case change hostname

hostnamectl set-hostname main.training.local

be sure to restart apache to take

systemctl restart httpd

```
### Variant 2: Remove <VirtualHost> </VirtualHost> - block vom ssl.conf
```

ssl.conf - Created by installation of mod_ssl.

cd /etc/httpd/conf.d/

vi ssl.conf

remove ....

systemctl restart httpd

```
### Use certbot to configure
```

certbot --apache --register-unsafely-without-email

```
### Test with your browser
```

https://ap1.t3isp.de

```
### Harden configuration using HSTS
```

https://de.wikipedia.org/wiki/HTTP_Strict_Transport_Security

/etc/httpd/conf.d/z_security.conf

Header set Strict-Transport-Security "max-age=31536000"

systemctl restart httpd

```
### Test Certificate with ssl labs

  * https://ssllabs.com

### Refs:

  * https://www.digitalocean.com/community/tutorials/how-to-secure-apache-with-let-s-encrypt-on-centos-8

### Client Base certificate


### Walkthrough (RHEL/Rocky)
```

mkdir -p /usr/local/src/SSL cd /usr/local/src/SSL

Schritt 1: ROOT-CA erstellen (firmenweit)

openssl genrsa -out ssl.netways.de_rootca.key 4096

Schritt 2: ROOT-CA Zertifikat erstellen

openssl req -x509 -new -nodes -key ssl.netways.de_rootca.key -sha256 -days 3650 -out ssl.netways.de_rootca.pem

Schritt 3: Key für Client erstellen

openssl genrsa -out ssl.netways.de_client1.key 4096

Schritt 4: CSR (Certificate Signing Request erstellen)

openssl req -new -key ssl.netways.de_client1.key -out ssl.netways.de_client1.csr

Schritt 5: Struktur für OpenSSL anlegen (default Verzeichnise, einkompiliert in openssl

mkdir -p demoCA/newcerts && mkdir demoCA/certs && mkdir demoCA/crl && echo 00 > demoCA/serial && touch demoCA/index.txt

Schritt 5a:

vi demoCA/openssl.cfg

SSLeay example configuration file.

This is mostly being used for generation of certificate requests.

RANDFILE = ./.rnd

[req] default_bits = 2048 default_keyfile = keySS.pem distinguished_name = req_distinguished_name encrypt_rsa_key = yes default_md = sha1

[req_distinguished_name] countryName = Country Name (2 letter code)

organizationName = Organization Name (eg, company)

commonName = Common Name (eg, YOUR name)

[ca] default_ca = CA_default # The default ca section

[CA_default]

dir = ./demoCA # Where everything is kept certs = $dir/certs # Where the issued certs are kept crl_dir = $dir/crl # Where the issued crl are kept database = $dir/index.txt # database index file. ##unique_subject = no # Set to 'no' to allow creation of # several certificates with same subject. new_certs_dir = $dir/newcerts # default place for new certs.

certificate = $dir/cacert.pem # The CA certificate serial = $dir/serial # The current serial number crl = $dir/crl.pem # The current CRL private_key = $dir/private/cakey.pem# The private key RANDFILE = $dir/private/.rand # private random number file

name_opt = ca_default # Subject Name options cert_opt = ca_default # Certificate field options

default_days = 365 # how long to certify for default_crl_days = 30 # how long before next CRL default_md = md5 # which md to use. preserve = no # keep passed DN ordering

policy = policy_anything

[policy_anything] countryName = optional stateOrProvinceName = optional localityName = optional organizationName = optional organizationalUnitName = optional commonName = supplied emailAddress = optional

Schritt 6: csr unterschreiben lassen

openssl ca -in ssl.netways.de_client1.csr -cert ssl.netways.de_rootca.pem -keyfile ssl.netways.de_rootca.key -out ssl.netways.de_client1.crt -days 3650 -config demoCA/openssl.cfg

Schritt 7: anderes Format erstellen für Import

openssl pkcs12 -export -in ssl.netways.de_client1.crt -inkey ssl.netways.de_client1.key -out NETWAYS_Client_gmimietz.p12

Schritt 8: ROOT-Ca kopiert

cp -a ssl.netways.de_rootca.pem /etc/pki/ca-trust/source/anchors/

Schritt 9:

put certificate into complete on in all pem-file

update-ca-trust extract update-ca-trust force-enable

Schritt 10: überprüft - common name from ca-root-certificate

cd /etc/pki/ca-trust/extracted/pem cat tls-ca-bundle.pem | grep ca.t3isp.de

Schritt 11 - vhost konfigurieren

Überprüft Client Zertifikat aktivieren

RHEL / Rocky

SSLCACertificateFile "/etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem" SSLVerifyClient require SSLVerifyDepth 5

Schrit 12 - Server neu starten

systemctl restart httpd

Schritt 13:

NETWAYS_Client_gmimietz.p12 - file auf Windows Rechner kopieren

z.B. base64 NETWAYS_Client_gmimietz.p12 > client.txt

cmd.exe

certutils -decode client.txt client.p12

Doppelt anklicken und importieren

Browser öffnen und Seite öffnen, jetzt man angemeldet.

Wenn Zertifikat, FEHLERMELDUNG

```
### Where to store the ca-cert (Different on RHEL and Debian)

  * https://stackoverflow.com/questions/37043442/how-to-add-certificate-authority-file-in-centos-7

### Refs:

  * https://www.netways.de/blog/series/ssl-leicht-gemacht/

## SELinux

### Neuen unbekannten Port freischalten, z.B. 86


### Walkthrough (kurzer Weg)
```

/etc/httpd/conf.d/httpd.conf

Weiteren port eintrag

Listen 86

systemctl restart httpd

serer startet nicht

semanage port -a -t http_port_t -p tcp 86 systemctl start httpd.service

```
### Walkthrough LÖsung - lange Form
```

/etc/httpd/conf.d/httpd.conf

Weiteren port eintrag

Listen 86

systemctl restart httpd

serer startet nicht

sealert muss installiert sein, wenn nicht

dnf whatprovides sealert

paket installieren

cd /var/log/audit sealert -a audit.log > report.txt

##.In der Report finden wir auch semanage anweisungen

In welcher Kategorie (Port-Kategorie) soll der Port hinzugefügt werden

Maßgeblich ist die Zeile, in der 80,81 etc vorkommt

semanage port -l | grep 80

http_port_t

```
### In welchem Kontext läuft der Apache-Server
```

ps auxZ | grep httpd ..... :httpd_t:s0

```
### Neues DocumentRoot verwenden
```

Schritt 1:

neuen Ordner anlegen

sudo su - mkdir /htdocs echo "testinhalt" > /htdocs/index.html

Schritt 2:

Zum 1. Debuggen SELinux deaktivieren

setenforce 0

Schritt 3:

änderungen in config /etc/httpd/conf/httpd.conf

DocumentRoot /htdocs

Directoryeintrag damit Zugriff erlaubt ist

Relax access to content within /htdocs.

<Directory "/htdocs"> AllowOverride None # Allow open access: Require all granted ##

Schritt 4: Reload und testn

systemctl reload httpd

Sollte funktionieren

curl -I http://192.168.56.103

Schritt 5: SELinux aktivieren und context setzen

setenforce 1 semanage fcontext -a -t httpd_sys_content_t '/htdocs/.*' restorecon -vr /htdocs

######## 2 things to consider

no perfect solution, because on relabeling (e.g. touch /.autorelabel and reboot)

it will be gone

1. chcon -R -t httpd_sys_content_t /htdocs/

2. '/htodcs/.' *is classic regex (0 or any chars =* .),

different to bash

echo .* -> show all hidden files: .*

#######

Schritt 6: Testing

now it works

```
## Absichern

### Standard-Apache-Seite deaktivieren (RHEL/Rocky)


### Walkthrough
```

for testing delete index.html

cd /var/www/html mv index.html index.html.bkup

cd /etc/httpd/conf.d mv welcome.conf welcome.conf.noshow # whatever suffix you like echo "# Disabled for security reasons" > welcome.conf

systemctl restart httpd

```
### Standard-Fehlerseiten konfigurieren


### Warum ?

  * Dem Angreifer so wenig wie möglich Informationen geben.
  * Prinzip: Datensparsamkeit

### Achtung
```

ohne Leerzeichen in den Hochkommas funktioniert nicht

Apache wirft einen Fehler

ErrorDocument 404 ""

```
### Wie ?
```

Variante Centos

z_security, damit es als letztes geladen wird und nicht

andere configs das auf der Serverebene überschreiben

cd /etc/httpd/conf.d/

vi z_security.conf

400 - Bad Request

ErrorDocument 400 " "

403 - Permission Denied

ErrorDocument 403 " "

404 - Not found

ErrorDocument 404 " "

500 Internal Server Error

ErrorDocument 500 " "

systemctl reload httpd

Try if you want

http://192.168.56.102/somepage-which-is-not-present.html

```
### Verzeichnislisting-autoindex deaktivieren/härten


### Variante 1: Ich benötige gar keine autoindex auf dem Server
```

Centos / RHEL / Rocky

Schritt 1:

cd /etc/httpd/conf.modules.d

vi 00-base.conf

Zeile deaktivieren - Kommentar davor

LoadModule autoindex_module modules/mod_autoindex.so

Schritt 2:

cd /etc/httpd/conf.d mv autoindex.conf autoindex.conf.noshow

Zur Sicherheit, damit kein Update das rüberbügelt

echo "# Disabled for security reasons" > autoindex.conf

Schritt 3:

systemctl restart httpd

Und testen wenn gewünscht

cd /var/www/html mv index.html index.html.noshow

Achtung: Welcome - Seite muss aktivieren siehe

Siehe Menüpunkt unter Sicherheit im Inhaltsverzeichnis dieses PDF'S

Jetzt darf kein Index mehr kommen

curl -I http://192.168.56.102/

```
### Variante 2: Ich schalte autoindex, aber will es evtl an anderer Stelle verwenden
```

z_ deshalb, weil sie als letztes geladen werden

/etc/httpd/conf.d/z_security.conf

Schritt 1

Security Measure 2

Geht nicht, weil durch <Directory - Eintrag

in /etc/httpd/conf/httpd.conf ueberschreiben

Options -Indexes

/var/www/html - spezifischer als

/var/www/

Ueberschreiben der Default-Seite

<Directory "/var/www/html"> Options -Indexes

Setzen fuer neue VirtualHost - Projekte

z.B. /var/www/meine-domain.de

<Directory "/var/www"> Options -Indexes

Schritt 2

systemctl restart httpd

evtl testen

cd /var/www/html mv index.html index.html.notused curl -I http://192.168.56.102

```
### Keine Apache Version kommunizieren
```

RHEL / Rocky

/etc/httpd/conf.d/z_security.conf ServerTokens Prod

systemctl restart httpd

Test

curl -I http://192.168.56.102

```
### Rechte härten
```

RHEL / Rocky

cd /var/www chown -R apache:apache html

Ausführungsrechte nur für Verzeichnisse,

keine Ausführsrechte für Dateien (erreicht mit X)

chmod -R u-x,g-x,u=rwX,g=rwX,o= daten

```
## Logs

### Journal persistent setzen
```

als root - wenn nicht vorhanden

wenn vorhanden ist es bereits persistent

mkdir /var/log/journal systemctl restart systemd-journal-flush

```
### Piped Logging


### Example
```

<VirtualHost *:80> ServerName casino.jochen.t3isp.de ServerAlias www.casino.jochen.t3isp.de

DocumentRoot /var/www/casino.bw.de/html ErrorLog /var/log/httpd/casino.bw.de-error.log CustomLog "|/usr/local/bin/logme.sh" combined

##/usr/local/bin/logme.sh ##!/bin/bash

while read line do echo "$line" >> /var/log/httpd/somenew.log done < "${1:-/dev/stdin}"

STEP 3: SET PERMISSIONS
chmod a+x /usr/local/bin/logme.sh

STEP 4: RESTART
systemctl restart httpd

```
### Refs:

  * https://httpd.apache.org/docs/2.4/logs.html#piped

## Proxy

### Reverse Proxy


### Einfacher Proxy
```

auf balancer

ProxyPass / http://10.135.0.9 ProxyPassReverse / http://10.135.0.9

Braucht der Proxy den Hosteintrag oder nicht ?

Default deaktiviert und in der Regel nicht benötigt

ProxyPreserveHost on

systemctl restart httpd

Achtung bei RHEL müssen unbedingt in SELinux ausgehende

Verbindungen erlaubt werden

Wenn selinux aktiviert ist -> sestatus -> enforcing

setsebool -P httpd_can_network_connect=1

```
### Balanced.
```

```
### Refs:

  * https://httpd.apache.org/docs/2.4/mod/mod_proxy.html#proxypreservehost

### Reverse Proxy Balancer


### Example
```

<Proxy "balancer://mycluster"> BalancerMember "http://10.135.0.9" BalancerMember "http://10.135.0.12"

important to set / twice in line

If you set it as last char in first column, and also has to be at the end of URL

DOES NOT WORK for sub pages: ProxyPass "/" "balancer://mysqlcluster"

USE THIS ->

ProxyPass "/" "balancer://mycluster/" ProxyPassReverse "/" "balancer://mycluster/"

```
### Sticky Session Id for php (sticky session explained)

  * http://www.markround.com/archives/33-Apache-mod_proxy-balancing-with-PHP-sticky-sessions.html

### References

  * https://httpd.apache.org/docs/2.4/mod/mod_proxy_balancer.html

## Tipps & Tricks

### In den Root-Benutzer wechseln
```

einloggen als normaler Benutzer z.B. benutzer: kurs (wenn ich unter kurs eingeloggt bin)

sudo su -

eingeben des Passworts des Benutzers

```
### Wo bin ich ?
```

1. Ich erkenne es am prompt (Beginn der Zeile)

pwd - Print working directory

pwd

```
### Nach Direktive suchen
```

grep -r "DirectoryIndex" /etc/httpd

oder

cd /etc/httpd grep -r "DirectoryIndex" .

```
### Praktische Ausgabe von langen Seiten - less
```

```
### Open a file with less
```

less /etc/services

Why ?

Leichtere Navigation

```
### Pipen mit less (ausgabe an less schicken)
```

ls -la | less cat /etc/services | less

```
### Suchen in less
```

##Innerhalb von less /suchbegriff + RETURN

nächstes Suchergebnis

n

```
###  Springen ans Ende/an den Anfang
```

Innerhalb von less

ans Ende

G

an den Anfang

1g

zu einer bestimmten Zeile (Zeile 5)

5g

```
### In die Hilfe rein
```

h

wieder raus

q

```
### cloudinit-script zum Anlegen von Server auf digitalocean


### Example Rocky/RHEL with Apache

  * Attention: better change password, currrently password
  * Tested on Rocky 8
```

##cloud-config users:

- name: 11trainingdo shell: /bin/bash

runcmd:

- sed -i "s/PasswordAuthentication no/PasswordAuthentication yes/g" /etc/ssh/sshd_config
- echo " " >> /etc/ssh/sshd_config
- echo "AllowUsers 11trainingdo" >> /etc/ssh/sshd_config
- echo "AllowUsers root" >> /etc/ssh/sshd_config
- systemctl reload sshd
- sed -i '/11trainingdo/c 11trainingdo:6res/PlXwgMx0Lunp$oRK58bWSJcy4ERNh.mZFTx3TfcYGWntPYPcihvUjIRQdhBv3E.OiHnEZn31ClDqtpVz7alPSY8NmTcbT8Swpn1:17476:0:999 /etc/shadow
- echo "11trainingdo ALL=(ALL) ALL" > /etc/sudoers.d/11trainingdo
- chmod 0440 /etc/sudoers.d/11trainingdo
- dnf install -y httpd
- systemctl start httpd
- systemctl enable httpd

Prozesse

Prozesse anzeigen - ps/pstree -p

Prozesse anzeigen

```
ps -ef
ps aux  # x alle Prozesse anzeigen, die nicht an ein Terminal gebunden sind
```

systemctl (läuft Dienst)

```
systemctl status sshd
```

Prozeßbaum anzeigen (meist nicht für die Praxis notwendig)

```
pstree -p
```

Prioritäten und NiceNess

nice aus dem userspace möglich

```
## userspace - ich führe als Benutzer ein Programm
## nice - Festlegen ein programm zum Kernel ist

## Priorität in top - pr
## je niedriger die Zahl für den Prozess in top ist, desto
## höher die Priorität
## höchste Priorität -100
-100 bis 39

## -100 = rt
## Statt -100 steht rt dort in top in der Spalte PR

## -59 höhe Priorität als 0 als 20 etc.

## Aus nice - Anfragen. von -20 -> +19 machen der Kernel
## in der Regel
n + 20 z.B. Niceness= -20. = -20 + 20 = 0
## D.h. höchte Priorität über nice im Kernel ist 0
## die niedrigste Priorität 39
```

Prozesse in Realtime ausführen wie geht ?

```
chrt # change realtime
## Welche Warteschlangen gibt es
chrt -m

## Packe top in die realtime Warteschlange und führe es aus
## --rr realtime roundrobin Warteschlange
## Führe top mit der Priorität -100 aus
chrt --rr 99 top
```

Ref:

- https://medium.com/@joseagustin.barra/understanding-priority-levels-in-linux-cd8c82eb4dd

Logs/Loganalyse

Logfile beobachten

```
## Terminal 1
tail -f /var/log/syslog

## Terminal 2 - write to logfile e.g.
logger meine_nachricht
```

Dienste debuggen

Walkthrough

```
## Dienst startet nicht / nach Ausführen von systemctl restart wird Fehlermeldung ausgegeben
systemctl restart httpd

## Schritt 1 : status -> was sagen die logs (letzte 10 Zeilen)
systemctl status httpd
```

```
## Nicht fündig-> Schritt 2:
jourrnalctl -xe

## Nicht fündig -> Schritt 3:
journalctl -u httpd.service

## Nicht fündig -> Schritt 4:
## Spezifisches Log von Dienst suchen
## und evtl. LogLevel von Dienst hochsetzen
## z.B. bei mariadb (durch Internetrecherche herausfinden)
less /var/log/httpd/error_log

## Nicht fündig -> Schritt 5
## Allgemeines Log
## Debian/Ubuntu
/var/log/syslog
## REdhat/Centos
/var/log/messages
```

Find error in logs quickly

```
cd /var/log/httpd
## -i = case insensitive // egal ob gross- oder kleingeschrieben
cat error_log | grep -i error
```

Journal analysieren

Show all boots

```
 journalctl --list-boots
 0 3c3cf780186642ae9741b3d3811e95da Tue 2020-11-24 14:29:44 CET<80><94>T>
lines 1-1/1 (END)
```

Show boot log

```
journalctl -b
```

Journal persistent

- Normalerweise (auf den meisten Systemen), überlebt das Journal kein Reboot

```
## persistent setzen
## Achtung: in /etc/systemd/journald.conf muss Storage=auto gesetzt sein
## Dies ist auch der Default - Fall
## Achtung Achtung: Alle gezeigten Einträge mit # am Anfang sind die Default-Werte (in journald.conf)
mkdir /var/log/journal
systemctl restart systemd-journal-flush.service
```

Restrict how much is logged / data

```
## in /etc/systemd/journald.conf
SystemMaxUse=1G
```

journalctl

```
## ubuntu
journalctl -u ssh
```

Show journalctl for specific Field

```
journalctl -o json-pretty
journalctl _PID=1 # show all entries for systemd - command startet as first program after kernel is loaded
journalctl _UID=120 # show all log entries for specific user
```

Dienste/Runlevel(Targets verwalten)

Die wichtigsten systemctl/service

Apache starten / stoppen / aktivieren

```
## Apache wird nach der Installation nicht standardmäßig gestartet in Centos/RHEL
systemctl status httpd
systemctl start httpd
systemctl status httpd
systemctl is-enabled httpd
```

```
## Rückgabewert des letzten Befehls
## Wenn nicht aktiviert kommt eine 1 raus
echo $?

systemctl enable httpd
systemctl is-enabled httpd
## Jetzt kommt eine 0 raus
echo $?
```

systemctl Beispiele

```
## Wie heisst der Dienst / welche Dienste gibt es ? (nur wenn der service aktiviert ist).
systemctl list-units -t service
## für apache
systemctl list-units -t service | grep ^httpd
## die Abkürzung
systemctl -t service | grep ^httpd

## Wie finde ich einen service, der noch nicht aktiviert ist ?
systemctl list-unit-files -t service | grep httpd

## Rebooten des Servers
## verweist auf systemctl
reboot
systemctl reboot
shutdown -r now

## Halt (ohne Strom ausschalten)
halt
systemctl halt
shutdown -h now

## Poweroff
poweroff
systemctl poweroff
```

Wie sehe ich, wie ein Service konfiguriert ist / Dienstekonfiguration anzeigen ?

```
## z.B. für Apache2
systemctl cat apache2.service
```

Wie kann ich rausfinden, wie die runlevel als targets heissen ?

```
cd /lib/systemd/system
root@ubuntu2004-104:/lib/systemd/system# ls -la run*target
lrwxrwxrwx 1 root root 15 Jan  6 20:47 runlevel0.target -> poweroff.target
lrwxrwxrwx 1 root root 13 Jan  6 20:47 runlevel1.target -> rescue.target
lrwxrwxrwx 1 root root 17 Jan  6 20:47 runlevel2.target -> multi-user.target
lrwxrwxrwx 1 root root 17 Jan  6 20:47 runlevel3.target -> multi-user.target
lrwxrwxrwx 1 root root 17 Jan  6 20:47 runlevel4.target -> multi-user.target
lrwxrwxrwx 1 root root 16 Jan  6 20:47 runlevel5.target -> graphical.target
lrwxrwxrwx 1 root root 13 Jan  6 20:47 runlevel6.target -> reboot.target
```

Welche Dienste sind aktiviert/deaktiviert

```
systemctl list-unit-files -t service
```

Dienste bearbeiten

```
systemctl edit sshd.service
## Dann eintragen
[Unit]
Description=Jochen's ssh-server
## Dann speichern und schliessen (Editor)

systemctl daemon-reload
systemctl status
```

Targets (wechseln und default)

```
## Default runlevel/target auslesen
systemctl get-default
## in target wechseln
systemctl isolate multi-user
```

```
## Default target setzen (nach start/reboot)
systemctl set-default multi-user
```

Alle Target anzeigen in die ich reinwechseln kann (isolate)

```
## Ubuntu
grep -r "AllowIsolate" /lib/systemd/system
/lib/systemd/system/reboot.target
...
...
...
systemctl isolate reboot.target
```

Dienste maskieren, so dass sie nicht gestartet werden können

```
systemctl mask apache2
## kann jetzt gestartet werden
systemctl start apache2

## de-maskieren
systemctl unmask apache2
## kann wieder gestaret werden
systemctl start apache2
```

systemctl Cheatsheet

- https://access.redhat.com/sites/default/files/attachments/12052018_systemd_6.pdf

Hilfe

Hilfe zu Befehlen

Möglichkeiten der Hilfe

```
## anhand von ps

vi -h
ps --help
man ps
info ps
```

-h oder --help --> eines geht immer

```
## Beispiel ls
ls -h # geht nicht für Hilfe
ls --help # geht !
```

Navigation in den man-pages

```
q - verlassen von man
Pfeil oben/unten
PageUp/PageDown
G # für ans Ende der Datei springe
1g # in die erste Zeile
```

Suche mit in man-pages

```
/Suchwort [Enter]
n # nächster Treffer (kleines n)
N # letzter Treffer
```

Firewall

firewalld

Install firewalld and restrict ufw

```
## Schritt 1: ufw deaktivieren
systemctl stop ufw
systemctl disable ufw
ufw disable # zur Sicherheit
ufw status
## -> disabled # this has to be the case

## Schritt 2: firewalld
apt install firewalld
systemctl start firewalld
```

```
systemctl enable firewalld
systemctl status firewalld
systemctl status ufw
```

Is firewalld running ?

```
## is it set to enabled ?
systemctl status firewalld
firewall-cmd --state
```

Command to control firewalld

- firewall-cmd

Best way to add a new rule

```
## Step1: do it persistent -> written to disk
firewall-cmd --add-port=82/tcp --permanent

## Step 2: + reload firewall
firewall-cmd --reload
```

Zones documentation

man firewalld.zones

Zones available

```
firewall-cmd --get-zones
block dmz drop external home internal public trusted work
```

Active Zones

```
firewall-cmd --get-active-zones
## in our case empty
```

Show information about all zones that are used

```
firewall-cmd --list-all
firewall-cmd --list-all-zones
```

Add Interface to Zone ~ Active Zone

```
firewall-cmd --zone=public --add-interface=enp0s3 --permanent
firewall-cmd --reload
firewall-cmd --get-active-zones
public
  interfaces: enp0s3
```

Default Zone

```
## if not specifically mentioned when using firewall-cmd
## .. add things to this zone
firewall-cmd --get-default-zone
public
```

Show services

```
firewall-cmd --get-services
```

Adding/Removing a service

```
firewall-cmd --permanent --zone=public --add-service=ssh
firewall-cmd --reload
firewall-cmd --permanent --zone=public --remove-service=ssh
firewall-cmd --reload
```

Add/Remove ports

```
## add port
firewall-cmd --add-port=82/tcp --zone=public --permanent
firewall-cmd --reload

## remove port
firewall-cmd --remove-port=82/tcp --zone=public --permanent
firewall-cmd --reload
```

Enable / Disabled icmp

```
firewall-cmd --get-icmptypes
## none present yet
firewall-cmd --zone=public --add-icmp-block-inversion --permanent
firewall-cmd --reload
```

Working with rich rules

```
## Documentation
## man firewalld.richlanguage

## throttle connectons
firewall-cmd --permanent --zone=public --add-rich-rule='rule family=ipv4 source address=10.0.50.10/32 service name=http log
level=notice prefix="firewalld rich rule INFO:   " limit value="100/h" accept'
firewall-cmd --reload #
firewall-cmd --zone=public --list-all

## port forwarding
firewall-cmd --get-active-zones
firewall-cmd --zone=public --list-all
firewall-cmd --permanent --zone=public --add-rich-rule='rule family=ipv4 source address=10.0.50.10 forward-port port=42343
protocol=tcp to-port=22'
firewall-cmd --reload
firewall-cmd --zone=public --list-all
firewall-cmd --remove-service=ssh --zone=public

##


## list only the rich rules
firewall-cmd --zone=public --list-rich-rules

## persist all runtime rules
firewall-cmd --runtime-to-permanent
```

References

- https://www.linuxjournal.com/content/understanding-firewalld-multi-zone-configurations#:~:text=Going%20line%20by%20line%20through,or%20source%20associated%20with%20it.
- https://www.answertopia.com/ubuntu/basic-ubuntu-firewall-configuration-with-firewalld/

Scannen und Überprüfen mit telnet/nmap

Netzwerk/Dienste

Hostname setzen

```
## please do it root
hostnamectl
hostnamectl set-hostname server1.training.local
## only reflects after new login
su -
```

Dokumentation

Apache module mit Direktiven

- https://httpd.apache.org/docs/2.4/en/mod/

Apache Konfigurationen (Directives) für alle Module

- https://httpd.apache.org/docs/2.4/mod/directives.html

Linux Security - PDF

- https://schulung.t3isp.de/documents/linux-security.pdf