

# PILOTPROJEKT FÜR DIE IMPLEMENTIERUNG EINES TRANSPARENTEN WEBPROXYS MIT BENUTZERAUTHENTIFIZIERUNG UND ZUWEISUNG DER WEBINHALTE

**Auszubildener**

Bruno Reboucas de Carvalho Penteadó

Voigtstrasse 4

04179 Leipzig

**Ausbildungsberuf**

Fachinformatiker-Systemintegration

**Bildungsträger**

Z&P Schulung GmbH

Rabensteinplatz 1

04179 Leipzig

**Pratikumsfirma**

Sage Software GmbH

Karl-Heine-Straße 109

04229 Leipzig

# Sommerprüfung 2016

## **Ausbildungsberuf**

Fachinformatiker/-in Systemintegration

## **Prüfungsbezirk**

Leipzig FISY 2 (T2, V1)

Herr Bruno Reboucas de Carvalho Penteado

Identnummer: 387569

E-Mail: bruno.reboucas@gmail.com, Telefon: +4915756840932

Ausbildungsbetrieb: Fachinformatiker Systemintegration

Projektbetreuer: Herr Christoph Petschnig

E-Mail: christoph.petschnig@sage.com, Telefon: +49 341 48440 3506

## **Thema der Projektarbeit**

Pilotprojekt für die Implementierung eines transparenten Webproxy mit  
Benutzerauthentifizierung und Zuweisung der Webinhalte.

# 1 Thema der Projektarbeit

Pilotprojekt für die Implementierung eines transparenten Webproxy mit Benutzerauthentifizierung und Zuweisung der Webinhalte.

## 2 Geplanter Bearbeitungszeitraum

Beginn: 15.03.2016

Ende: 20.05.2016

## 3 Projektbeschreibung

Bei der Entwicklungsabteilung meiner Praktikumsfirma hatte bisher jeder Entwickler eine eigene lokale Testumgebung mit lokalen Intranetdiensten. Dieser Zustand war weder effizient noch zentral administrierbar. Aus diesen Gründen soll ein Pilotprojekt Klarheit darüber bringen, ob und wie eine zentrale und intelligente Einrichtung der individuellen Testumgebungen mittels Webproxy und erforderlichen und skalierbaren Diensten durchgeführt werden kann. Hierbei soll eine personenabhängige Authentifizierung am Webproxy erfolgen. Durch die Authentifizierung entscheidet der Webproxy nach einem Berechtigungsplan über die Berechtigungen, Zeiten und Zuweisung der Intranetdienste für die Entwickler und soll die Loginstatistiken erfassen.

Ein zukünftiges Projekt könnte dieses Pilotprojekt im Rahmen eines Loadbalancers weiter verwenden.

## 4 Projektumfeld

Bei dem Pilotprojekt handelt es sich um ein betriebsinternes Produkt für das interne Verwaltungssystemprogramm um mit der öffentlichen Homepage abzugleichen. Der Auftraggeber ist somit die Firma Sage selbst. Eine andersweitige Verwendung ist nicht geplant.

## 5 Projektphasen mit Zeitplanung

Planung 6,0 Std.

- Kundengespräch und Erstellung des IST-Zustands  
1,0 Std.
- Erstellung des Sollzustand  
3,0 Std.
- Erstellung des Ablaufplans  
2,0 Std.

Durchführung (gesamt) 16,0 Std.

- Installation von Servers und testen Clients

3,0 Std	
- Konfiguration von Server mit elastisch skalierbaren Container	
8,0 Std	
- Konfiguration von Transparenten Proxy Server	
5,0 Std	
Test und Übergabe	6,0 Std.
- Test zwischen Verbindungen Server und Clients über Proxy	
3,5 Std.	
- Erstellung der administrativen Kundendokumentation	
1,5 Std.	
- Übergabe an den Kunden	
1,0 Std.	
Erstellung der Dokumentation	7,0 Std.
Benötigte Gesamtzeit	35 Std.

## 6 Dokumentation zur Projektarbeit

Der Projektdokumentation werden folgende Dokumente angehängt, um Vollständigkeit zu gewährleisten:

- Lastenheft
- Pflichtenheft
- Netzwerkkonfiguration/dokumentation
- Benutzerdokumentation
- Administratordokumentation

## 7 Anlagen

keine

## 8 Präsentationsmittel

Eigener Laptop (Rechner)  
Beamer (VGA/HDMI)

## 9 Hinweis!

Ich bestätige, dass der Projektantrag dem Ausbildungsbetrieb vorgelegt und vom Ausbildenden genehmigt wurde. Der Projektantrag enthält keine Betriebsgeheimnisse. Soweit diese für die Antragstellung notwendig sind, wurden nach Rücksprache mit dem Ausbildenden die entsprechenden Stellen unkenntlich gemacht.

Mit dem Absenden des Projektantrages bestätige ich weiterhin, dass der Antrag eigenständig von mir angefertigt wurde. Ferner sichere ich zu, dass im Projektantrag personenbezogene Daten (d. h. Daten über die eine Person identifizierbar oder bestimmbar ist) nur verwendet werden, wenn die betroffene Person hierin eingewilligt hat.

Bei meiner ersten Anmeldung im Online-Portal wurde ich darauf hingewiesen, dass meine Arbeit bei Täuschungshandlungen bzw. Ordnungsverstößen mit „null“ Punkten bewertet werden kann. Ich bin weiter darüber aufgeklärt worden, dass dies auch dann gilt, wenn festgestellt wird, dass meine Arbeit im Ganzen oder zu Teilen mit der eines anderen Prüfungsteilnehmers übereinstimmt. Es ist mir bewusst, dass Kontrollen durchgeführt werden.

genehmigt



## INHALTSVERZEICHNIS

1. Ausgangslage.....	1
1.1 Projektbeschreibung.....	1
1.2 Gründe für den Pilotprojektauftrag .....	1
1.3 Projektziele .....	1
1.4 Projektumfeld .....	1
1.5 Projektschnittstellen .....	1
1.5.1 Personelle Schnittstellen .....	1
1.5.2 Technische Schnittstellen .....	1
1.6 Projekt Aufgaben .....	1
2. Projektplanung .....	2
2.1 Ist-Zustand .....	2
2.2 Soll-Zustand .....	2
2.3 Anforderungsanalysen .....	2
2.3 Entscheidungsplanung .....	2
2.3.1 Hardware vs. Software .....	2
2.3.2 Squid vs. Webserver .....	3
2.3.3 Apache vs. NginX .....	4
2.3.4 Graphik vs. Script.....	4
2.4 Netzwerkplanung.....	5
2.5 Ressourcenplanung.....	5
2.5.1 Softwareplanung .....	5
2.5.2 Betriebssysteme die zum Einsatz kommen .....	6
2.5.3 Hardwareplanung.....	6
2.6 Kostenplanung.....	6
2.5.1 Betreutes Stunden Kosten.....	6
2.5.2 Hardware Kosten .....	6
2.5.3 Software Kosten .....	7
2.5.4 Wirtschaftkeitanalyse .....	7
2.7 Zeit- und Ablaufplanung .....	7
3. Projektdurchführung.....	7
3.1 Betriebssysteme Installation .....	7
3.1.1 Test Clients Betriebssystems Installationen .....	7
3.1.2 Test Server Betriebssystems Installationen.....	8
3.1.3 Server Betriebssystems Installationen .....	8
3.1.4 Container Installationen .....	9
3.2 System Konfigurationen.....	9
3.2.1 Test Clients Konfigurationen .....	9

3.2.2 Test Server Konfigurationen .....	10
3.2.3 Imitation Server Konfiguration .....	10
3.2.4 Container Server Konfiguration .....	11
3.2.5 Container Konfigurationen .....	11
3.3 Proxy System.....	11
3.3.1 Domain Name Server (DNS) .....	11
3.3.2 Benutzer Authentifizierung .....	12
3.3.3 Webserver als Proxy .....	12
3.3.4 Webserver als Proxy mit Lastenausgleich (Load-Balancing).....	13
3.3.5 Webalizer (Logdatei) .....	14
3.3.6 Routing bis zum Client Netzwerk.....	15
3.3.7 Proxy Firewall .....	15
4. Testphase .....	15
4.1 Clients Automatische IP-Adresse Tests (DHCP) .....	15
4.2 Clients Namen Auflösung Tests .....	16
4.2.1 Imitation Server Namen Auflösung Tests .....	16
4.2.2 Proxy Server Namen Auflösung Tests.....	16
4.2.3 Testumgebung Namen Auflösung Tests .....	16
4.3 Verbindung Tests .....	16
4.3.1 Verbindung zur Imitation Server Tests .....	17
4.3.2 Verbindung zur Proxy Server Tests.....	17
4.3.3 Verbindung zur Testumgebung Tests .....	17
4.5 Sicherheitstests .....	17
4.5.1 Proxy Server Verbindung Sicherheitstests .....	17
4.5.2 Testumgebung Verbindung Sicherheitstests .....	17
4.5.3 Testumgebung Benutzer Authentifizierung Sicherheitstests.....	17
4.6 Load-Balancing Tests .....	18
4.6.1 Clients Load-Balancing Test Skripte.....	18
4.6.2 Ausgleich Tests .....	18
5. Übergabephase .....	18
5.1 Übergabe zum Praktikumsfirma .....	18
6. Projektauswertung.....	19
6.1 Qualitätsforderungen .....	19
Einhaltung der ProjektStandards .....	19
6.2 Soll und ist Zeit abgelaufen von Projekt .....	19
6.2 Erweiterbarkeit .....	19
7. Fazit .....	19
7.1 Projekt erfolgreich .....	19



7.2 Ausblick .....	19
8. Kundendokumentation .....	19
9. Admin-Dokumentation .....	19
Glossar .....	20
ACL(Access Control List).....	20
Atom Editor.....	20
Container .....	20
CoreOS .....	21
Debian.....	21
DHCP (Dynamic Host Configuration Protocol) .....	21
DMZ (Demilitarized Zone).....	21
dig .....	22
DNS (Domain Name Server).....	22
dnsdomainname .....	23
Docker.....	23
Firewall .....	23
GitHub.....	23
hostname.....	24
IDE (Integrated Development Environment) .....	24
GUI (Grafische Benutzeroberfläche).....	24
IP-Adresse (Internet Protocol Address) .....	24
ipconfig .....	25
ISO-Abbild .....	25
Linux.....	25
Linux Mint .....	25
Logdatei .....	25
MAC-Adresse (Media-Access-Control) .....	25
MAC-Adresse (Media-Access-Control) .....	26
nmap.....	26
nslookup .....	26
Portverteilung.....	26
Proxy .....	26
Plugins.....	26
Squid .....	26
Ubuntu.....	27
Ubuntu Mate .....	27
Virtual Box .....	27
Transparent-Proxy .....	27

Vagrant .....	27
Webalizer.....	27
whois.....	28
Windows 7 .....	28
ANLAGES .....	28
Anlage 1: Netzwerkplanung.....	28
Anlage 2: Gantt-Diagramm .....	29
Anlage 3: Tabelle Soll-Ist Zeitableufen.....	31
Konfigurationen Anlagen .....	32
Konfiguration Anlage 1: Test Client 1 Konfigurationen .....	32
Client 1 Name .....	32
Client 1 Bekannten Hosts .....	32
Client 1 Domainname .....	32
Client 1 Namensauflösung .....	32
Client 1 Netzwerkschnittstelle.....	32
Konfiguration Anlage 2: Test Client 2 Konfigurationen .....	33
Client 2 Name .....	33
Client 2 Bekannten Hosts .....	33
Client 2 Domainname .....	33
Client 2 Namensauflösung .....	33
Client 2 Netzwerkschnittstelle.....	33
Konfiguration Anlage 3: Test Client 3 Konfigurationen .....	34
Client 3 Name .....	34
Client 3 Netzwerkkonfigurationen .....	34
Konfiguration Anlage 4: Test Server 1 Konfigurationen .....	35
Test Server 1 Name .....	35
Test Server 1 Bekannten Hosts.....	35
Test Server 1 Netzwerkschnittstellen.....	35
Test Server 1 Apache Webserver Ordner Inhalt.....	36
Konfiguration Anlage 5: Test Server 2 Konfigurationen .....	36
Test Server 2 Name .....	36
Test Server 2 Bekannten Hosts.....	36
Test Server 2 Netzwerkschnittstellen.....	36
Test Server 2 Apache Webserver Ordner Inhalt.....	37
Konfiguration Anlage 6: Test Server 3 Konfigurationen .....	37
Test Server 3 Name .....	37
Test Server 3 Bekannten Hosts.....	37
Test Server 3 Netzwerkschnittstellen.....	37

Test Server 3 Apache Webserver Ordner Inhalt .....	38
Konfiguration Anlage 7: Imitation Server Konfigurationen .....	38
Imitation Server Name .....	38
Imitation Server Bekannten Hosts.....	38
Imitation Server Netzwerkschnittstellen .....	38
Imitation Server Portverteilung.....	39
Imitation Server DNS Konfigurationen .....	39
Imitation Server DHCP Konfigurationen .....	43
Konfiguration Anlage 8: Container Server Konfigurationen .....	44
Vagrant File .....	44
Konfiguration Anlage 9: Container 1 Konfigurationen .....	48
NginX Ordner Inhalt.....	48
Docker Datei Inhalt.....	48
Konfiguration Anlage 10: Container 2 Konfigurationen .....	48
NginX Ordner Inhalt.....	48
Docker Datei Inhalt.....	48
Konfiguration Anlage 11: Container 3 Konfigurationen .....	48
NginX Ordner Inhalt.....	48
Docker Datei Inhalt.....	48
Proxy System Anlagen .....	49
Proxy Anlage 1: Proxy-Server DNS Konfigurationen .....	49
Proxy Server rndc.key .....	49
Proxy Server db.webproxy.local.forward datei .....	50
Proxy Server db.webproxy.local.rev datei .....	51
Proxy Server named.conf datei .....	51
Proxy Server named.conf.local datei .....	52
Proxy Server named.conf.options datei .....	52
Proxy Anlage 2: Proxy-Server Benutzerrichtlinien Konfigurationen .....	54
Einsetzen neue Benutzer ins Benutzerrichtlinien .....	54
Inhalt htacces Datei ausdrückt nach neue Benutzer Einsetzung .....	54
Benutzer Löschung .....	54
Inhalt htacces Datei ausdrückt nach Benutzer Löschung .....	54
Proxy Anlage 3: Webserver als Proxy Beispiel Konfiguration .....	54
Proxy Anlage 4: Round-Robin Konfiguration .....	55
Proxy Anlage 5: Weighted Konfiguration.....	55
Proxy Anlage 6: Persistent Konfiguration .....	55
Proxy Anlage 7: Least-Connected Konfiguration .....	56
Proxy Anlage 8: Webalizer Webseite Konfiguration .....	56

Proxy Anlage 9: Firewall.....	57
Datei .....	57
Funktionen .....	57
Proxy Anlage 10: Proxy System.....	66
Datei .....	66
Flussdiagram.....	67
Funktionen .....	67
Tests Anlagen .....	98
Test Anlage 1: Clients IP-Adresse Tests .....	98
Client 1 IP-Adresse Tests .....	98
Client 2 IP-Adresse Tests .....	99
Client 3 IP-Adresse Tests .....	99
Test Anlage 2: Namen Auflösung von Imitation Server Tests.....	100
Client 1 Namen Auflösung von Imitation Server Tests.....	100
Client 2 Namen Auflösung von Imitation Server Tests.....	101
Client 3 Namen Auflösung von Imitation Server Tests.....	102
Test Anlage 3: Clients Namen Auflösung von Proxy Server Tests.....	102
Client 1 Namen Auflösung von Proxy Server Tests.....	102
Client 2 Namen Auflösung von Proxy Server Tests.....	103
Client 3 Namen Auflösung von Proxy Server Tests.....	103
Test Anlage 4: Clients Namen Auflösung von Testumgebung Tests .....	103
Client 1 Namen Auflösung von Testumgebung Tests.....	103
Client 2 Namen Auflösung von Testumgebung Tests.....	104
Client 3 Namen Auflösung von Testumgebung Tests.....	105
Test Anlage 5: Clients Verbindung Tests.....	105
Verbindung zu Imitation Server Tests .....	105
Verbindung zu Proxy Server Tests .....	105
Verbindung zu Testumgebung Tests .....	106
Test Anlage 6: Sicherheitstests.....	106
Proxy Server Netzwerdienst .....	106
Proxy Server Sicherheitstests.....	107
Testumgebung Sicherheitstests .....	112
Test Anlage 7: Client Load-Balancing Tests.....	115
Clients Load-Balancing Test Skripte.....	115
Round-Robing Load-Balancing Tests .....	116
Weighted Load-Balancing Tests .....	117
Persistent Load-Balancing Tests .....	117
Least-Connected Load-Balancing Tests .....	118

Kundendokumentation Anlage .....	119
Testumgebung .....	119
Einloggen .....	119
Nicht Autorisiert .....	119
Web Inhalt nach Authorisierung .....	120
Webalizer .....	120
Einloggen .....	120
Nicht Autorisiert .....	120
Webalizer anwendung nach Authorisierung .....	121
Admin-Dokumentation Anlage .....	122
1 Benutzer.....	122
1.1 Benutzer Einsetzen (Optionen 2).....	122
1.2 Benutzern Ausdrücken (Optionen 1).....	122
1.3 Existierten Benutzer Aktualisieren (Optionen 3) .....	123
1.4 Benutzer Löschen (Optionen 4).....	123
1.5 Benutzern Ausdrücken (Optionen 1).....	124
2 Servers .....	124
2.1 Server Einsetzen (Optionen 2).....	124
2.2 Server Ausdrücken (Optionen 1) .....	125
2.3 Server Löschen (Optionen 3) .....	126
3 Containers.....	127
3.1 Containers Einsetzen (Optionen 2).....	127
3.2 Containers Ausdrücken (Optionen 1) .....	129
3.3 Containers Löschen (Optionen 3) .....	129
Literaturverzeichnis.....	132

## 1. AUSGANGSLAGE

Im Folgenden soll ein Überblick über die Ausgangslage des Pilotprojektes geschaffen werden. Hierbei werden das Projekt und alle erforderlichen Teilaufgaben genau definiert, sowie das Projektumfeld und alle Schnittstellen beschreiben.

### 1.1 PROJEKTBESCHREIBUNG

Bei der Entwicklungsabteilung meiner Praktikumsfirma hatte bisher **jeder Entwickler eine eigene lokale Testumgebung mit lokalen Intranetdiensten**. Dieser Zustand war weder effizient noch zentral administrierbar. Aus diesen Gründen soll ein Pilotprojekt Klarheit darüber bringen, ob und wie eine zentrale und intelligente Einrichtung der individuellen Testumgebungen mittels Webproxy und erforderlichen und skalierbaren Diensten durchgeführt werden kann. Hierbei soll eine personenabhängige Authentifizierung am Webproxy erfolgen. Durch die Authentifizierung entscheidet der Webproxy nach einem Berechtigungsplan über die Berechtigungen, Zeiten und Zuweisung der Intranetdienste für die Entwickler und soll die Login Statistiken erfassen.

### 1.2 GRÜNDE FÜR DEN PILOTPROJEKTAUFTRAG

Für die Entwicklungsabteilung wird eine Testumgebung erzeugt um Änderungen der Testkonfigurationen in kürzester Zeit einzusehen und wenn möglich zu verändern und um diese den gewünschten Bedürfnissen anzupassen.

### 1.3 PROJEKTZIELE

Ziel des Pilotprojekts ist es, den Entwicklern eine schnelle und skalierbare Testumgebung bereit zu stellen um Änderungen bzw. Anpassungen an den Systemkonfigurationen beliebig testen zu können.

### 1.4 PROJEKTUMFELD

Bei dem Pilotprojekt handelt es sich um eine betriebsinterne Lösung für das interne Systemverwaltungsprogramm, damit die fehlende Benutzerauthentifizierung eingeführt werden kann. Der Auftraggeber ist somit die Praktikumsfirma selbst. Eine anderweitige Verwendung ist nicht geplant.

### 1.5 PROJEKTSCHNITTSTELLEN

An der Realisierung des Pilotprojekts arbeiten keinen anderen Mitarbeiter oder Kollegen der Praktikumsfirma mit.

Außer das ich bei Fragen in engen Kontakt mit meinem Vorgesetzten stehe.

---

#### 1.5.1 PERSONELLE SCHNITTSTELLEN

Herr XXXX ist der Ansprechpartner für dieses Pilotprojekt.

---

#### 1.5.2 TECHNISCHE SCHNITTSTELLEN

Die Netzwerkverbindung wird nicht über das Netzwerk der Praktikumsfirma geregelt sondern wird über eine Virtualisierung erzeugt. Da aus sicherheitstechnischen Gründen keine Verbindung über das Praktikumsnetzwerk aufgebaut werden soll.

### 1.6 PROJEKT AUFGABEN

Aufbau einer virtuellen Netzwerkverbindung zwischen den Clients zu einer Testumgebung über den Webproxy.

## 2. PROJEKTPLANUNG

Im diesem Punkt wird die genaue Planung des Pilotprojektes mit Übersichten veranschaulicht. Hierzu zählen neben Analyse von Ist- und Soll-Zustand auch Zeit- und Ablauf, Entscheidungsfindung, Ressourcen- und Kostenplanung.

### 2.1 IST-ZUSTAND

Bisher haben die Entwickler ihre Tests lokal auf den Workstations mit den hierfür installierten Intranetdiensten ausgeführt. Damit war es nicht möglich geänderte Konfigurationen mit anderen Entwicklern im Netzwerk zu teilen.

### 2.2 SOLL-ZUSTAND

In Zukunft sollen die lokalen Testumgebungen auf mehrere Server verteilt werden. Der Zugriff zu diesen Servern soll skalierbar für beliebige Testumgebungen und Entwickler mit einer geplanten Benutzerrichtlinie realisiert werden. Bei der Nutzung dieser Lösung soll das Verteilungssystem nach dem Aufruf des Servernamens (FQDN) eine Loginmöglichkeit anbieten, die die Entwickler nach erfolgreicher Authentifizierung zu diesem Server weitergeleiten soll.

Für dieses Projekt darf der physische Server der Praktikums Firma nicht eingesetzt werden. Da aber für dieses Projekt die DNS und DHCP Konfigurationen benötigt werden, werden die Konfigurations Dateien vom Physischen Server auf einen Virtuellen Server verschoben und für die Tests des Pilotprojekts verwendet, da die Entwicklungsabteilung den Physischen Server für ihre Arbeiten und Test benötigt und als Imitation Server genannt.

Das Verteilungssystem soll die Möglichkeit bieten Belastungstests der (Server-)Auslastung der Testumgebung mittels Load-Balancing durchzuführen, deren Konfiguration den Entwicklern unterliegt.

Die Konfiguration der Testumgebung und, einschließlich der Zugriffsberechtigung, erfolgt auf den Servern und ist nicht Bestandteil dieses Projektes.

### 2.3 ANFORDERUNGSANALYSEN

Ziel des Pilotprojekts ist es, einfach machen für den Entwickler schneller, skalierbar, problemlos eine Test Umgebung einzusetzen.

- Zentralisierter Zugriff auf der Testumgebung mittels Authentifizierung
- Plattform-/systemübergreifend
- Skalierbar Clients- und Testumgebungen
- Load-Balancing
- Absicherung durch Firewall
- Kostengünstig/Lizenzfrei
- Geeigneter Proxy Server
- Geeignete Webserver

### 2.3 ENTSCHEIDUNGSPLANUNG

Im Folgenden werden Abwägungen dargestellt, die zur Entscheidungsfindung herangezogen wurden, auf welchem System das Pilotprojekt aufgebaut wurde

---

#### 2.3.1 HARDWARE VS. SOFTWARE

Für die Lösung und Ermittlung der geeigneten Hard- oder Software wurde eine Entscheidungsmatrix erstellt.

---

## KOSTEN

Da in diesem Projekt kaum Kosten anfallen sollen, wird die Bewertung mit einem Prozentsatz von 60% bewertet und verrechnet.

---

## FLEXIBILITÄT

Da die Flexibilität in diesem Projekt kaum eine Rolle spielt, wird dies mit einem Prozentsatz von 20% verrechnet.

---

## ERWEITERBARKEIT

Die Erweiterung des Systems und vorhandenen Komponenten werden mit einem Prozentsatz von 30% verrechnet.

---

### TABELLE HARDWARE VS. SOFTWARE

	Bewertung	Hardware	Software	Gewichtete Hardware	Gewichtete Software
Kosten	60%	0	5	0	3,0
Flexibilität	20%	2	4	0,4	0,8
Erweiterbarkeit	30%	1	5	0,3	1,5
<b>SUMME</b>	100%	3	14	0,7	<b>5,3</b>

---

### TABELLE HARDWARE VS. SOFTWARE ERGEBNIS

Anhand der Auswertung oben genannter Tabelle wurde die Entscheidung zu Gunsten einer Softwarelösung getroffen.

---

## 2.3.2 SQUID VS. WEBSERVER

Für die Lösung und Ermittlung der geeigneten Software von Squid und Webserver wurde eine Entscheidungsmatrix erstellt.

---

### BENUTZER LOGIN

Das Benutzer Login wird mit 30% berechnet, da gewünscht wird, dass man die Kenntnis darüber hat wer eingeloggt war und ist.

---

### ARBEITSSPEICHER CACHE

Der Arbeitsspeicher Cache wird mit 30% berechnet, da dieser schneller ist und genügend Pufferzeit bietet.

---

### VERSCHLÜSSLUNG

Die Verschlüsselung wird mit 20% bewertet, da die Sicherheit einen wichtigen Aspekt in der Datenverarbeitung und Integrität spielt.

---

### LOAD-BALANCING MÖGLICHKEIT

Das Load-Balancing wird mit 30% bewertet, da die Lastenverteilung für eine bessere Systemauslastung und Verteilung sorgt.

---

### TABELLE SQUID VS. WEB

	Bewertung	Squid	Web	Gewichtete Squid	Gewichtete Web
Benutzer Login	30%	0	5	0	1,5



RAM Cache	30%	0	5	0	1,5
SSL	20%	5	3	1	0,6
Load-Balancing	20%	4	4	0,8	0,8
<b>SUMME</b>	<b>100%</b>	<b>9</b>	<b>17</b>	<b>1,8</b>	<b>4,4</b>

#### TABELLE SQUID VS. WEB ERGEBNIS

Anhand der Auswertung oben genannter Tabelle wurde die Entscheidung zu Gunsten einer Webserverlösung getroffen.

#### 2.3.3 APACHE VS. NGINX

In der Gegenüberstellung wurde ermittelt welche der bessere Webserver ist.

##### PLUGINS

Die erweiterten benutzbaren Plugins werden mit 30% Berechnet.

##### LEICHTGEWICHTS

Das Leichtgewicht wird mit 30% Berechnet, da diese nicht so viel Hardwareleistung benötigt.

##### IMPLEMENTIERUNG

Die Implementierung wird mit als 30% Berechnet, da die Implantierung und Konfigurierung auf die die Hardware setzt.

##### ORDNERSTRUKTUR

Die Ordnerstruktur wird mit 30% Berechnet, da die Ordnerstruktur im System übersichtlich angeordnet ist.

#### TABELLE APACHE VS. NGINX

	Bewertung	Apache	NginX	Gewichtete Apache	Gewichtete NginX
Plugins	30%	4	4	1,2	1,2
Leichtgewichts	40%	3	4	1,2	1,6
Implementierung	15%	3	4	0,45	0,6
Ordnerstruktur	25%	2	4	0,5	1
<b>SUMME</b>	<b>100%</b>	<b>12</b>	<b>16</b>	<b>3,55</b>	<b>4,4</b>

#### TABELLE APACHE VS. NGINX ERGEBNIS

Anhand der Auswertung oben genannter Tabelle wurde die Entscheidung zu Gunsten einer NginX-webserverlösung getroffen.

#### 2.3.4 GRAPHIK VS. SCRIPT

Hier wird einschneidet auf welche Lösung benutzt würde für der Proxy Server zur der ganze System aktualisieren.

---

## KEINE ERWEITERE DIENST

Für einen leichteren und schnelleren Server ist es nötig, unnötige Dienste auszuschalten. Die wird Berechnet mit 60%

---

## GRAFISCHE BENUTZEROBERFLÄCHE (GUI)

Das System wird meistens von den Systemadministratoren und Entwicklungsabteilung für eine kurze Zeit benutzt. Solche Benutzern haben Große Affinität mit Systemen ohne GUI arbeiten, deswegen ist die Bewertung für GUI mit 10% Berechnet wurden.

---

## FIREWALL AKTUALISIERUNG

Die Aktualisierung der Firewall im vorhandenen System wird mit 15% Berechnet.

---

## BENUTZER AKTUALISIERUNG

Die Systemaktualisierung für die Benutzer Aktualisierung wird mit 15% Berechnet.

---

## TABELLE GRAPHIK VS. TERMINAL

	Bewertung	Graphik	Terminal	Gewichtete Graphik	Gewichtete Terminal
Keine erweiterer Dienst	60%	1	5	0,6	3
GUI	10%	5	2	0,5	0,2
Firewall Aktualisierung	15%	4	4	0,6	0,6
Benutzer Aktualisierung	15%	4	4	0,6	0,6
<b>SUMME</b>	100%	14	15	2,3	<b>4,4</b>

---

## TABELLE GRAPHIK VS. TERMINAL ERGEBNIS

Anhand der Auswertung oben genannter Tabelle wurde die Entscheidung zu Gunsten einer Terminallösung getroffen.

## 2.4 NETZWERKPLANUNG

Nach der Entscheidung wurde ein dementsprechender Netzwerkplan erstellt.

Sieht [Anlage 1: Netzwerkplanung](#) für weitere Details.

## 2.5 RESSOURCENPLANUNG

Wie besprochen mit dem Herr XXX der mich bei diesem Projekt betreut, sollen die Kosten so gering wie möglich gehalten werden.

---

### 2.5.1 SOFTWAREPLANUNG

---

#### 2.5.1.1 VIRTUALISIERUNGSSOFTWARE

##### [VAGRANT](#)

Eine Vagrant System wird eingesetzt um eine maximale Skalierbarkeit und um Ressourcen zu erhalten. Da dieser für den Container Server dient.

##### [VIRTUALBOX](#)

Alle anderen Maschinen wurden Virtuell in der Virtualbox eingesetzt, um eine Graphische Darstellung zu haben.

---

## INTEGRIERTE ENTWICKLUNGSUMGEBUNG (IDE)

Für die Entwicklung der Skripte wurde für die Benutzung [Atom Editor](#) IDE von [GitHub](#) verwendet.

---

## 2.5.2 BETRIEBSSYSTEME DIE ZUM EINSATZ KOMMEN

---

### 2.5.2.1 LINUX SYSTEMEN

---

#### [UBUNTU](#) SERVER 14.04.4 LTS

---

Für den Proxy und Imitation Server wurde ein Ubuntu Server 14.04. LTS Installiert.

---

#### [UBUNTU MATE](#) 15.10

---

Als Testsystem wurde ein Ubuntu Mate 15.10 Installiert.

---

#### [LINUX MINT](#) 17.3 (ROSA)

---

Als Testsystem wurde ein Linux Mint 17.3 Betriebssystem installiert.

---

#### [DEBIAN](#) 8.3.0

---

Als weiteres Testsystem wurde ein Debian 8.3.0 als Testserver installiert.

---

#### [COREOS](#)

---

Das CoreOS Betriebssystem wird für den Einsatz für den Container Server installiert.

---

### 2.5.2.2 WINDOWS SYSTEME

---

Auf Wunsch der Praktikumsfirma, wurde für Testzwecke ein [Windows 7](#) Betriebssystem Installiert.

---

## 2.5.3 HARDWAREPLANUNG

---

Baut auf die vorhandene Hardware auf.

---

## 2.6 KOSTENPLANUNG

---

Die Kosten die für dieses Projekt werden in Form von der Arbeitszeit ermittelt die für das Pilotprojekt benötigt wurde.

---

### 2.5.1 BETREUTES STUNDEN KOSTEN

---

#### 2.5.1.1 MINDESTLOHN

---

Mindestlohn pro Stunde von Herr XXX beträgt 50,00€

---

#### 2.5.1.2 BETREUER ZEITVERBRAUCH

---

Für die Unterstützung stand mir der Herr XXX für 4 Stunden zur Seite, und betreute mich bei diesem Projekt.

---

#### 2.5.1.3 TABELLE

---

Praktikumsfirma Mitarbeiter	Total Stunden	Stundenkosten
Herr XXX	4	50,00€
<b>SUMME</b>		<b>200,00€</b>

---

### 2.5.2 HARDWARE KOSTEN

---

In den folgenden Punkten wird erklärt, dass die Kosten für die zukünftigen Möglichkeiten auf einen Server installiert, eingesetzt und benutzt werden können.

---

#### 2.5.2.1 SERVER

Der Server HP-ProLiant DL380 G5 mit Xeon E5410 QuadCore 2.33 GHz und 8 GB RAM kostet 119,99 €

---

#### 2.5.2.2 FESTPLATTE

Die Festplatte mit einer Größe von 500GB kostet 20,00€

---

#### 2.5.2.3 HARDWARE KOSTEN TABELLE

Beschreibung	Stückzahl	Ges. Kosten
Server HP-ProLiant DL380	1	119,99€
Festplatte	2	20,00€
<b>SUMME</b>		<b>159,99€</b>

---

#### 2.5.3 SOFTWARE KOSTEN

Softwares sind Lizenz frei und Kostenlos.

---

#### 2.5.4 WIRTSCHAFTSKEITANALYSE

Das Projekt soll maßgeblich den Administrationsaufwand im Bereich der Testumgebungen reduzieren und massiv vereinfachen sowie die Lastenverteilung der zentralisierten Zugriffe auf diese Testumgebungen günstig abwickeln, um die zuständigen Server optimal auszulasten.

Ferner soll der Arbeitsplatz der Entwickler redundant sein, da sich lokal keine Testumgebung mehr befinden soll. Auch soll die Erstellung der Testumgebung vereinfacht und effizient erstellt werden können.

Durch diese Maßnahmen sollen deutliche Einsparungen erreicht werden, die sich in der Zukunft zeigen werden.

---

### 2.7 ZEIT- UND ABLAUFPLANUNG

Die Zeitablaufplanung wurde mit dem Programm Microsoft-Project erstellt und ist als Anlage zur Einsicht hinterlegt.

Siehe [Anlage 2: Gantt-Diagramm](#) für das Gantt-Diagramm.

---

## 3. PROJEKTDURCHFÜHRUNG

In den folgenden Punkten wird die Durchführung der einzelnen Arbeitsschritte genauer erklärt.

---

### 3.1 BETRIEBSSYSTEME INSTALLATION

---

#### 3.1.1 TEST CLIENTS BETRIEBSSYSTEMS INSTALLATIONEN

Auf den Tests Clients wurden verschiedene Systeme installiert.

- Test Client 1
  - Linux Mint 17.3 (Rosa) wurde installiert
- Test Client 2
  - Ubuntu Mate 15.10 wurde installiert.
- Test Client 3
  - Zum Vergleich wurde ein Windows System installiert. Die gewünschte Version ist Windows 7.

Die Installation von verschiedenen Diensten auf den Linux Clients ist notwendig um diese später bei Tests durchzuführen.

```
sudo apt-get install whois nmap traceroute curl
```

---

### 3.1.2 TEST SERVER BETRIEBSSYSTEMS INSTALLATIONEN

Als Test Server wurde ein einziges mal ein Debian 8.3.0 installiert und geklont. Dieser wurde mit verschiedenen [MAC-Adressen](#) vergeben.

---

### 3.1.3 SERVER BETRIEBSSYSTEMS INSTALLATIONEN

---

#### 3.1.3.1 IMITATION SERVER INSTALLATIONEN

Der Imitation Server ist ein Virtueller Server der einen Physischen Server nach empfunden ist, und die gleichen Namensauflösung (DNS) und IP-Adresse verteiler (DHCP) Konfigurationsdateien enthält wie der Physische Server.

Auf dem Imitation Server wird der Ubuntu Server 14.04.03 Installiert.

Die folgenden Dienste werden auf dem Imitation Server Installiert:

- **DNS**
  - Wird es installiert für Namensauflösung.
- **DHCP**
  - Wird es installiert für IP-Adresse verteiler für die Clients.

---

#### 3.1.3.2 PROXY SERVER INSTALLATIONEN

Auf den Proxy Server wird der Ubuntu Server 14.04.03 Installiert.

Die folgenden Dienste werden auf dem Imitation Server Installiert:

- **DNS**
  - Es wurde installiert für verterleiten der Namensauflösung für den Testumgebung.
- **NginX**
  - Der NginX wurde eingesetzt für den Proxy-Dienst.
- **Webalizer**
  - Wurde installiert um Logideatein besser lesen zu können.

---

#### 3.1.3.3 CONTAINER SERVER INSTALLATIONEN

Hier wird erklärt wie eine erfolgreiche Vagrant als Container Server über eine Virtuelle Maschine eingesetzt wird.

- **Vagrant Abbild von GitHub herunterladen**
  - Ein Abbild vom CoreOS System wurde von GitHub heruntergeladen.
    - `git clone https://github.com/coreos/coreos-vagrant/`
- **System Variablen Einsetzen**
  - Für die Installation von dem System werden die Systemvariablen eingesetzt.
    - `DISCOVERY_TOKEN`: wurde es benutzt für zugreifen auf den Vagrant Addbild Server einzugreifen.
      - `DISCOVERY_TOKEN=`curl -s https://discovery.etcd.io/new` && perl -p -e "s@#discovery: https://discovery.etcd.io/<token>@discovery: $DISCOVERY_TOKEN@g" user-data.sample > user-data`
    - `NUM_INSTANCES`: einsetzung der nummer von Vagrant Maschinen auf den System.

- `export NUM_INSTANCES=1`
- **Add SSH-Key auf den Vagrant Konfigurationen**
  - Für eine sichere Verbindung mit der Virtuell-Maschine wurde ein benötigter SSH-Key eingesetzt.
    - `ssh-add ~/.vagrant.d/insecure_private_key`
- **Vagrant Abbild Aktualisierung mit installation**
  - Es wurde eine Aktualisierung und Installation des Container Servers unter Nutzung des Vagrant CoreOS Abbildes vorgenommen.
    - `vagrant box update`

---

### 3.1.4 CONTAINER INSTALLATIONEN

Auf den Containern wurde ein Abbild von einem Webserver Container installiert. Der gewünschte Webserver für die Container basierte auf NginX.

- **Container 1**
  - Ordner Einsetzen: hier wir eingestellt der ordner die verknüpfen auf den Container Mount mit die Webinhalt.
    - `core@core-01 ~ $ mkdir /nginx_servers/ng1`
  - Neue Container von NginX Container-Abbildung und verteilt auf den Port 4001
    - `core@core-01 ~ $ docker create -p 4001:80 --name ng1 -v ~/nginx_servers/ng1:/usr/share/nginx/html:ro nginx`
- **Container 2**
  - Ordner Einsetzen: hier wir eingestellt der ordner die verknüpfen auf den Container Mount mit die Webinhalt.
    - `core@core-01 ~ $ mkdir /nginx_servers/ng2`
  - Neue Container von NginX Container-Abbildung und verteilt auf den Port 4002.
    - `core@core-01 ~ $ docker create -p 4002:80 --name ng2 -v ~/nginx_servers/ng2:/usr/share/nginx/html:ro nginx`
- **Container 3**
  - Ordner Einsetzen: hier wir eingestellt der ordner die verknüpfen auf den Container Mount mit die Webinhalt.
    - `core@core-01 ~ $ mkdir /nginx_servers/ng3`
  - Neue Container von NginX Container-Abbildung und verteilt auf den Port 4003.
    - `core@core-01 ~ $ docker create -p 4003:80 --name ng3 -v ~/nginx_servers/ng3:/usr/share/nginx/html:ro nginx`

## 3.2 SYSTEM KONFIGURATIONEN

---

### 3.2.1 TEST CLIENTS KONFIGURATIONEN

Auf den Test Clients wurde diese Konfigurationen realisiert:

- **Linux Clients**
  - **Client Name**
    - Bei diesem Punkt wird der Computernamen für den Test Clients eingesetzt.
  - **Bekannte Hosts**
    - Bei diesem Punkt wird die Datei für das Betriebssystem für die lokale Namensauflösung für die Test Clients eingesetzt.
  - **Domainname**
    - Bei diesem Punkt wird die Datei für das Betriebssystem für den Domainnamen für die Test Clients eingesetzt.
  - **Namensauflösung**

- Bei diesem Punkt wird die Datei automatisch geschrieben, nachdem eine IP-Adresse für die Test Clients vergeben wurde.
  - **Netzwerkschnittstelle**
    - Bei diesem Punkt wird erklärt wie die Test Clients seine IP-Adresse vom Imitation Server durch DHCP erhält.
- **Windows Clients**
  - **Client Name**
    - Bei diesem Punkt wird der Computernamen für den Test Client 3 eingesetzt.
  - **Netzwerkkonfigurationen**
    - Bei diesem Punkt wird die Netzwerkkonfigurationen für eine Windowsclient für den Test Client 3 eingesetzt.

Siehe [Konfiguration Anlage 1: Test Client 1 Konfiguration](#) für weitere Details.

Siehe [Konfiguration Anlage 2: Test Client 2 Konfiguration](#) für weitere Details.

Siehe [Konfiguration Anlage 3: Test Client 3 Konfiguration](#) für weitere Details.

---

### 3.2.2 TEST SERVER KONFIGURATIONEN

- Auf den Test Clients wurde diese Konfigurationen realisiert:
- **Servername**
  - Einsetzen der Konfigurationsdatei für den Servernamen auf die Test Server.
- **Bekannte Hosts**
  - Einsetzen der Konfigurationsdatei für den bekannten Host auf die Test Server.
- **Netzwerkschnittstellen**
  - Hier werden die IP-Adressen der Netzwerkschnittstellen vom den Test Server zugewiesen.
- **Apache Webserver**
  - Bei diesem Punkt wird der Webserver-Standard-Webpage-Inhalt vom den Test Server eingestellt.

Siehe [Konfiguration Anlage 4: Test Server 1 Konfiguration](#) für weitere Details.

Siehe [Konfiguration Anlage 5: Test Server 2 Konfiguration](#) für weitere Details.

Siehe [Konfiguration Anlage 6: Test Server 3 Konfiguration](#) für weitere Details.

---

### 3.2.3 IMITATION SERVER KONFIGURATION

Auf dem Imitation Server wurde diese Konfigurationen realisiert:

- **Server Name**
  - Einsetzen der Konfigurationsdatei für den Servername Inhalt.
- **Bekannte Hosts**
  - Einsetzen der Konfigurationsdatei für den Host Inhalt.
- **Netzwerkschnittstellen**
  - Hier werden die IP-Adressen den Netzwerkschnittstellen zugewiesen.
- **Portverteilung**
  - Hier wird erklärt wie die Portverteilung über die Netzwerkschnittstellen bereitgestellt wird.
- **DNS (Domain Name Server)**
  - Die Clients verbinden sich mit dem Imitation-Server, da dieser die Namensauflösung im Netzwerk bereitstellt. Die Namensauflösung der Testumgebung wird über DNS-Forwarding durch den Proxy Server Namensauflösungsdienst für die Test-Server auf der Testumgebung im Netzwerk bereitgestellt.

- **rndc-Schlüssel**
  - Der Zweck dieser Signatur Transaktionen mit BIND dient der Authentifizierung. Somit kann der DHCP-Server die Domain name.local und webproxy.local aktualisieren, wenn er diesen Schlüssel verliert.
- **Datei: /etc/bind/db.name.local**
  - Hier wird die Konfiguration von den Clients im Netzwerk Host-Name eingesetzt.
- **Datei: /etc/bind/named.conf**
  - Hier wird die rndc-key Konfiguration eingesetzt.
- **Datei: /etc/bind/named.conf.local**
  - Hier wird die Konfiguration der lokalen Netzwerke vom Server unterstützt.
- **Datei: /etc/bind/named.conf.options**
  - Hier wird die Weiterleitung bis Proxy Server DNS eingesetzt.
- **DHCP (Dynamic Host Configuration Protocol)**
  - Hier wird allen Clients die auf der Liste stehen eine IP-Adresse vergeben. Clients deren Mac-Adresse nicht in der Liste stehen erhalten auch keine IP-Adresse. Somit ist eine Auflösung am Proxy-Server nicht möglich.

Die erforderliche Firewall ist ein externes Projekt.

Siehe [Konfiguration Anlage 7: Imitation Server Konfiguration](#) für weitere Details.

---

### 3.2.4 CONTAINER SERVER KONFIGURATION

Die heruntergeladene Vagrant-Datei wird für die Nutzung des Virtuellen Netzwerks auf die gewünschte Konfiguration gesetzt, damit die Virtuelle Maschine die Generierung fehlerfrei ausführen kann.

Siehe [Konfiguration Anlage 8: Container Server Konfiguration](#) für weitere Details.

---

### 3.2.5 CONTAINER KONFIGURATIONEN

Auf dem Container-Server ist eine Verknüpfung zum NginX Ordner. In diesem Ordner befindet sich die Konfigurationsdatei.

- **NginX Ordner Inhalt**
  - Auf dem Nginx Ordner Inhalt wird eine Verknüpfung zum Container gesetzt.
- **Docker Datei Inhalt**
  - Hier wird die Docker Datei eingesetzt, die ein Abbild vom Container benutzt und eine Verknüpfung zum Verzeichnis beinhaltet.

Für die kompletten Container 1 Konfigurationen siehe [Konfiguration Anlage 9: Container 1 Konfigurationen](#)

Für die kompletten Container 2 Konfigurationen siehe [Konfiguration Anlage 10: Container 2 Konfigurationen](#)

Für die kompletten Container 3 Konfigurationen siehe [Konfiguration Anlage 11: Container 3 Konfigurationen](#)

## 3.3 PROXY SYSTEM

In den folgenden Punkten wird der Einsatz des Proxy System über den Webserver genauer erklärt.

---

### 3.3.1 DOMAIN NAME SERVER (DNS)

Für die Notwendigkeit und Verwendung des Namensauflösungsdienstes (DNS) auf dem Proxy-Server dient die Funktionalität des Transparent-Proxy.



Der DNS-Dienst vom Imitation-Server dient für den Proxy-Server als Weiterleitung (DNS-Forward), sodass auf diese Weise ein Update auf der Testumgebung mit einer Einfügung an einem Server stattfinden kann. Somit ist eine Aktualisierung auf dem Imitation Server überflüssig.

Für die komplette DNS Konfiguration siehe [Proxy Anlage 1: Proxy-Server DNS Konfiguration](#).

- **rndc-Schlüssel**
  - Hier wird der Schlüssel für die Authentifizierung eingesetzt.
- **Datei: /etc/bind/db.webproxy.local.forward**
  - Hier wird die Forward-Konfiguration vom Server der Testumgebung für den Host-Name eingesetzt.
- **Datei: /etc/bind/db.webproxy.local.rev**
  - Hier wird die Reverse-Konfiguration vom Server der Testumgebung für den Host-Name eingesetzt.
- **Datei: /etc/bind/named.conf**
  - Hier wird die *rndc-key* Konfiguration eingesetzt.
- **Datei: /etc/bind/named.conf.local**
  - Hier wird die Konfiguration der lokalen Netzwerke vom Server unterstützt.
- **Datei: /etc/bind/named.conf.options**
  - Hier wird die [ACL\(Access Control List\)](#) für die Verschlüsselung auf dem DNS Dienst eingetragen.

---

### 3.3.2 BENUTZER AUTHENTIFIZIERUNG

Für den Zugriff auf die Testumgebung werden Authentifizierung eingerichtet, sodass nur autorisierte Benutzer Zugriff haben.

Die Benutzung von *htaccess* wird hier für den sicheren Zugriff auf die Testumgebung eingesetzt.

Zur Anwendung von Benutzerrichtlinien auf dem NginX Webserver sollte die Webpagekonfigurationsdatei folgende Anweisungen beinhalten:

- Welche Art von Authentifizierung für den Zugriff benutzt wird.
  - ***auth\_basic "Restricted";***
- Die Benutzerrichtliniendatei mit den Benutzern, die Zugriff zur Testumgebung haben.
  - Beispiel: ***auth\_basic\_user\_file /etc/nginx/htaccess/.htpasswd;***

Für die komplette Benutzer Authentifizierung Konfiguration siehe [Proxy Anlage 2: Proxy-Server Benutzer Authentifizierung Konfiguration](#).

---

### 3.3.3 WEBSERVER ALS PROXY

Der Transparent-Proxy wird eingesetzt. Es kann ein Container oder ein normaler Server sein. Der Proxy dient der Weiterleitung und unterscheidet nicht zwischen diesen beiden.

Für die Beispielkonfiguration siehe [Proxy Anlage 3: Webserver als Proxy Beispiel Konfiguration](#).

Zum Einsetzen des NginX Webserver für einen Transparent-Proxy muss die Webpagekonfigurationsdatei folgende Konfigurationen beinhalten:

- Von welcher Adresse und Port sollte der Web-Server die Verbinden erlauben
  - Beispiel: ***listen 172.16.2.11:80;***
- Die Webpage Adresse
  - Beispiel: ***server\_name single.webproxy.local;***
- Art des Proxy-Severs

- **proxy\_redirect** default;
- von welcher Adresse und Port soll umgeleitet werden
  - Beispiel: **proxy\_pass** <http://10.0.0.11:80>;

---

### 3.3.4 WEBSERVER ALS PROXY MIT LASTENAUSGLEICH (LOAD-BALANCING)

Für den Lastenausgleich des Nginx Dienstes wird die Upstream Konfiguration hinzugefügt, die für den Server als Ausgleichsverbindung dient.

Für die *location* Konfiguration mit der Variable **proxy\_set\_header** wird der Proxy Server als Transparent-Proxy verwendet. Mit der Variable **proxy\_pass** erfolgt die Verwendung der Upstream-Umfrage.

```
location / {

    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

    proxy_pass http://<UPSTREAM_POOL_NAME>;

}
```

---

#### 3.3.4.1 ROUND-ROBIN

Round-Robin ist die Standardmethode, die NginX verwendet, weil diese notwendig ist für die Upstream-Umfrage Einstellung

```
upstream <ROUND-ROBIN-UPSTREAM-NAME> {

    server <SERVER_1-ADRESSE:PORT>;

    server <SERVER_2-ADRESSE:PORT>;

}
```

Für die Konfigurationsdatei siehe [Proxy Anlage 4: Round-Robin Konfiguration](#).

---

#### 3.3.4.2 WEIGHTED

Wenn die Gewichtungparameter für einen Server angegeben werden, wird die Wichtung als Teil der Lastausgleichsentscheidung berücksichtigt.

```
upstream <WEIGHTED-UPSTREAM-NAME> {

    server <SERVER_1-ADRESSE:PORT> weight=3;

    server <SERVER_2-ADRESSE:PORT>;

    server <SERVER_3-ADRESSE:PORT>;

}
```

Mit dieser Beispielkonfiguration werden alle 5 neuen Anfragen über die Anwendungsinstanzen wie folgt verteilt: 3-Anfragen an den Server-1, eine Anfrage an Server-2, und eine an Server-3.

Ebenso ist es in der neueren Versionen von NginX möglich, Wichtungen zu verwenden, um die am wenigsten verbundenen IP-Hash-Belastungen auszubalancieren. (NginX, 2016)

Für die Konfigurationsdatei siehe [Proxy Anlage 5: Weighted Konfiguration](#).

---

#### 3.3.4.3 PERSISTENT

Mit IP-Hash und der IP-Adresse des Clients wird ein Hashing-Schlüssel verwendet, um zu bestimmen, welcher Server in einer Server-Gruppe sein sollte, um die Anfragen der Clients auszuwählen. Dieses Verfahren stellt sicher, dass die Anfragen vom selben Client immer auf den gleichen Server ausgerichtet werden. Wenn dieser Server nicht mehr verfügbar ist, wird ein neuer Hashing-Schlüssel mit einem verfügbaren Server verwendet.

Für eine Persistent Load-Balancing Konfiguration muss die Variable *ip\_hash* in den Upstream-Umfrage-Einstellungen eingefügt werden.

```
upstream <PERSISTENT-UPSTREAM-NAME> {  
  
    ip_hash;  
  
    server <SERVER_1-ADRESSE:PORT>;  
  
    server <SERVER_2-ADRESSE:PORT>;  
  
}
```

Für die Konfigurationsdatei siehe [Proxy Anlage 6: Persistent Konfiguration](#).

---

#### 3.3.4.4 LEAST-CONNECTED

Least-Connected ermöglicht es situationsabhängig die Last auf verschiedene Anwendungsinstanzen umzuverteilen, wenn einige der Anfragen länger dauern.

Mit dem am Least-Connected Lastenausgleich, wird NginX nicht versuchen, einen besetzten Anwendungsserver mit übermäßigen Anfragen zu überlasten, sondern die neuen Anforderungen an einen weniger ausgelasteten Server verteilen.

Für eine Least-Connected Load-Balancing Konfiguration muss die Variable *least\_conn* in den Upstream-Umfrage-Einstellungen eingefügt werden.

```
upstream <LEAST-CONNECTED-UPSTREAM-NAME> {  
  
    least_conn;  
  
    server <SERVER_1-ADRESSE:PORT>;  
  
    server <SERVER_2-ADRESSE:PORT>;  
  
}
```

Für die Konfigurationsdatei siehe [Proxy Anlage 7: Least-Connected Konfiguration](#).

---

#### 3.3.5 WEBALIZER (LOGDATEI)

Die Log-Dateien können mit dem Webalizer Dienst für eine visuelle Darstellung und ein besseres Verständnis der Nutzung des Systems verwendet werden.

Der Webalizer kann nur von Benutzern verwendet werden, die die Rechte durch die Benutzerrichtlinien haben. Dies kann in Nginx durch Verwendung des htaccess-Prozess in der Webseite-Konfiguration von Webalizer erreicht werden.

Für die Aktualisierung der Logdatei ins Webalizer kann mit einem *webalizer* Kommandobefehl auf dem Proxy Server erreicht werden.

Für die Webalizer Konfiguration siehe [Proxy Anlage 8: Webalizer Webseite Konfiguration](#).

---

### 3.3.6 ROUTING BIS ZUM CLIENT NETZWERK

Der Proxy sollte Kenntnis darüber haben, welche dem vorhandenen Clients sich in einem anderen Netzwerk befinden.

- Auto Enable
  - Das Routing startet sobald das Netzwerk startet.  
*root@proxy:~# cat /etc/network/if-up.d/routingtabelle*  
*#!/bin/bash*  
*### ROUTING*  
*route add -net 192.168.3.0/24 eth1*
- Route Tabelle
  - Der Zugriff sollte durch die ETH1 Schnittstellen gehen.

*root@proxy:~# route -n*

*Kernel-IP-Routentabelle*

<i>Ziel</i>	<i>Router</i>	<i>Genmask</i>	<i>Flags</i>	<i>Metric</i>	<i>Ref</i>	<i>Use</i>	<i>Iface</i>
<i>0.0.0.0</i>	<i>192.168.1.1</i>	<i>0.0.0.0</i>	<i>UG</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>eth0</i>
<i>10.0.0.0</i>	<i>0.0.0.0</i>	<i>255.255.255.0</i>	<i>U</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>eth2</i>
<i>172.16.2.0</i>	<i>0.0.0.0</i>	<i>255.255.255.0</i>	<i>U</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>eth1</i>
<i>192.168.1.0</i>	<i>0.0.0.0</i>	<i>255.255.255.0</i>	<i>U</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>eth0</i>
<b><i>192.168.3.0</i></b>	<b><i>0.0.0.0</i></b>	<b><i>255.255.255.0</i></b>	<b><i>U</i></b>	<b><i>0</i></b>	<b><i>0</i></b>	<b><i>0</i></b>	<b><i>eth1</i></b>

---

### 3.3.7 PROXY FIREWALL

Die iptables-basierte Firewall am am Webproxy hat die Aufgabe nur Clients aus dem vom DHCP aufgebauten Netz zu erlauben. Damit sind andere Zugriffe nicht möglich.

Für die Firewall Konfiguration siehe [Proxy Anlage 9: Firewall](#).

## 4. TESTPHASE

### 4.1 CLIENTS AUTOMATISCHE IP-ADRESSE TESTS (DHCP)

Nach dem die Clients eingeschalten wurden, haben diese umgehend eine IP-Adresse vom Imitaion Server erhalten.

Die Clients 1,3 haben ohne Probleme eine IP-Adresse erhalten. Da Client 2 nicht auf der Liste steht hat dieser Client auch keine IP-Adresse vom DHCP Server erhalten.

Für die Linux-Clients werden diese Befehle zum Testen der Netzwerkkonfiguration verwendet.

- **hostname -l**
  - Bei dem hostname -l Kommandozeilenprogramm unter Unix ist ausgedrückt den Rechnernamen des Systems
- **dnsdomainname**
  - Bei der dnsdomainname Kommandozeilenprogramm unter Unix ist ausgedrückt den DNS-Domain-Namen des Systems
- **ip addr show**

- Bei der `ip addr show` Kommandozeilenprogramm unter Unix ist ausgedrückt die Konfiguration von IP-Netzwerkschnittstellen.

Für die Windows-Clients werden die Befehle zum Testen der Netzwerkkonfiguration verwendet

- **ipconfig**
  - Bei der Terminalbefehl `ipconfig` auf der Windows System die Konfiguration von IP-Netzwerkschnittstellen ausgedrückt wurde.

Für das Ergebnis der Tests siehe [Test Anlage 1: Clients IP-Adresse Tests](#)

## 4.2 CLIENTS NAMEN AUFLÖSUNG TESTS

Hier wird die Funktionalität der Namensauflösung beschrieben. Und ob eine Verbindung und Namensauflösung zum Server stattfinden kann.

Für die Linux-Clients werden diese Befehle zum Testen der Namensauflösung verwendet.

- **nslookup**
  - Bei dem Befehl `nslookup` auf Linux Systemen wird angezeigt wann der output vom Name bis IP-Adresse funktioniert.
- **dig**
  - Bei dem Befehl `dig` auf Linux Systemen werden die ganzen Informationen angezeigt die wir von dem DNS Server bekommen haben.
- **whois**
  - Bei diesem Befehl wird die Information des Domainbesitzers angezeigt, und man erhält Informationen über den Eigentümer, seine Kontaktinformationen und den Nameserver der Domain. Bei diesem Pilotprojekt handelt es sich um Beispielnamen.

Für die Windows-Clients werden die Befehle zum Testen der Namensauflösung verwendet

- **nslookup**
  - Bei dem Befehl `nslookup` auf Windows System wird angezeigt wann der output vom Name bis IP-Adresse funktioniert.

Für den Ergebnis von Client 1 siehe [Test Anlage 2: Clients Namen Auflösung Tests](#)

---

### 4.2.1 IMITATION SERVER NAMEN AUFLÖSUNG TESTS

Sobald die Clients eine IP-Adresse bekommen haben sollte automatisch die Namensauflösung zum DNS-Server stattfinden.

---

### 4.2.2 PROXY SERVER NAMEN AUFLÖSUNG TESTS

Sobald die Clients eine IP-Adresse bekommen haben wird automatisch eine Namensauflösung gemacht. Diese wird durch den Proxy-Server zum Imitation-Server hergestellt.

---

### 4.2.3 TESTUMGEBUNG NAMEN AUFLÖSUNG TESTS

Sobald die Clients alle eine IP-Adresse haben wird eine Auflösung zu den Servern per DNS-Forwarding hergestellt.

## 4.3 VERBINDUNG TESTS

Mit den Verbindungs Tests wurde sichergestellt, dass alle Clients eine Verbindung zum Server haben.

Für den Tests Ergebnis siehe [Test Anlage 5: Clients Verbindung Tests](#)

---

#### 4.3.1 VERBINDUNG ZUR IMITATION SERVER TESTS

Die Verbindungstests für die Clients wurden ausgeführt, um zu sehen ob sie wirklich eine Verbindung zum Imitation Server haben. Diese wird mit dem *tracroute* Terminal-Befehl erfolgen.

---

#### 4.3.2 VERBINDUNG ZUR PROXY SERVER TESTS

Die Verbindungstests für die Clients wurden ausgeführt, um zu sehen ob sie wirklich eine Verbindung zum Proxy Server haben. Diese wird mit dem *tracroute* Terminal-Befehl erfolgen.

Diese Tests können nur dann ausgeführt werden, wenn die Firewall auf dem Proxy Server deaktiviert ist.

---

#### 4.3.3 VERBINDUNG ZUR TESTUMGEBUNG TESTS

Hier noch einmal der *tracroute* Terminal-Befehl der verwendet wird, um die Clients Verbindungen zu testen.

Diese Tests können nur dann ausgeführt werden, wenn die Firewall auf dem Proxy Server deaktiviert ist.

### 4.5 SICHERHEITSTESTS

Die Sicherheitstests werden getestet, wenn der Server nur Verbindungen innerhalb des Netzwerks des Kunden, und nur auf speziellen Diensten ermöglichen kann.

Für die Tests Ergebnis siehe [Test Anlage 6: Sicherheitstests](#).

---

#### 4.5.1 PROXY SERVER VERBINDUNG SICHERHEITSTESTS

Auf allen Clients auf denen nmap installiert wurde um den Netzwerkscanner zu benutzen.

- **Ohne Firewall**
  - Mit diesem Test kann die ganze Datei des Servers angezeigt werden, und das diese ungeschützt ist.
  - Wenn die Tests ohne Firewall laufen, ist man in der Lage zu sehen, dass die SSH und DNS-Dienste geöffnet sind. Auch der gesamte TCP-Fingerprint und welche Art von Betriebssystem verwendet wird, ist zu sehen.
- **Mit Firewall**
  - Durch die eingeschaltete Firewall kann man die Hops auslesen, die einem die Informationen anzeigen dass die Firewall eingeschaltet ist.
  - Man erhält auch die Angabe dass die Verbindung für SSH und DNS-Dienst geschützt sind.

---

#### 4.5.2 TESTUMGEBUNG VERBINDUNG SICHERHEITSTESTS

Die Testumgebung mit den Servern haben keine Firewall. Alle Verbindungen werden durch den Proxyserver geschützt.

- **Ohne Firewall**
  - Es ist möglich dass der SSH-Dienst und Webserver ausgelesen werden können.
- **Mit Firewall**
  - Nur der Webserver kann ausgelesen werden.

---

#### 4.5.3 TESTUMGEBUNG BENUTZER AUTHENTIFIZIERUNG SICHERHEITSTESTS

Bei der Benutzer Authentifizierung nicht erfolgt, und es wird dem Benutzer nicht erlaubt die Verbindung in der Testumgebung.

- Verbindung Tests ohne Authentifizierung
  - Sobald der Versuch stattfinden sich zu Authentifizieren wird eine Fehlermeldung angezeigt.

```
web@cli1:~$ curl single.webproxy.local

<html>

<head><title>401 Authorization Required</title></head>

<body bgcolor="white">

<center><h1>401 Authorization Required</h1></center>

<hr><center>nginx/1.4.6 (Ubuntu)</center>

</body>

</html>
```

- Benutzer Verbindung mit erfolgt Authentifizierung
  - Sobald der Benutzer sich Authentifizieren kann wird eine Umleitung stattgegeben.

```
web@cli1:~$ curl <USER_LOGGING>:<USER_PASSWD>@single.webproxy.local
apache 1
```

## 4.6 LOAD-BALANCING TESTS

Der Load-Balancing-Test zeigt die Antwortreaktion vom Serveran der für alle Arten von Ausgleich testet.

Für die Test Ergebnis siehe [Test Anlage 7: Load-Balancing Tests](#).

### 4.6.1 CLIENTS LOAD-BALANCING TEST SKRIPT

Dieses Skript wird verwendet, um die Verbindung vom Kunden mit dem Load-Balancing verfahren zu testen.

### 4.6.2 AUSGLEICH TESTS

- Round-Robin Tests
  - Die Reaktion der Round-Robin-Tests sollte vom ersten bis zum letzten der Server-Umfrage-Liste rotieren.
- Weighted Tests
  - Die Reaktion aus den Weighted Tests sollten von den ersten rotierenden, bis zum letzten der Server-Umfrage-Liste mit den Gewichten angepasst werden.
- Persistent Tests
  - Die Reaktion der Persistent Tests soll Antworten darüber geben das nur mit dem ersten Server der Client verbunden war.
- Least-Connected Tests
  - Die Reaktion der Least-Connected Tests sollten vom ersten bis zum letzten der Server-Umfrage mit den eingestellten Antworten aus der Umfrage-Liste Reaktionszeit rotieren.

## 5. ÜBERGABEPHASE

### 5.1 ÜBERGABE ZUM PRAKTIKUMSFIRMA

Nach erfolgreichen Tests konnte das Projekt den Zuständigen übergeben und vorgeführt werden. In diesem Rahmen wurde die gewünschte Funktionalität gezeigt und eine entsprechende Einweisung

vorgenommen.

Da es keine Beanstandung gab, wurde das Projekt abgenommen und als abgeschlossen erklärt.

## 6. PROJEKTAUSWERTUNG

### 6.1 QUALITÄTSFORDERUNGEN

Zur Anforderung gehörte prinzipiell die Lastenverteilung bei den Authentifikationen zu den Testumgebungen. Diese wurde vollumgänglich erfüllt.

---

### EINHALTUNG DER PROJEKTSTANDARDS

Hierzu zählten die alle Aspekte aus dem Lastenheft, die ebenfalls erfüllt wurden. Alle Funktionen des Projekts und deren Standards wurden erfüllt.

### 6.2 SOLL UND IST ZEIT ABGELAUFEN VON PROJEKT

Ich habe mehr Zeit für das Projekt benötigt, da der Gesamtumfang des Projekts unterschätzt habe. Erschwerend kamen meine sprachlichen Defizite hinzu.

Siehe [Anlage 3: Tabelle Soll-Ist Zeitablaufen](#)

### 6.2 ERWEITERBARKEIT

Der Proxy-Server könnte auch für Zugriffe aus anderen Netzwerken konfiguriert werden ebenso kann der Loadbalancer hierfür angepasst werden. Ferner könne beliebig viele Testumgebungen ins System integriert werden. Hier kann von einer Vollskalierbarkeit gesprochen werden.

## 7. FAZIT

### 7.1 PROJEKT ERFOLGREICH

Mein persönliches Fazit ist, dass das vorliegende Projekt für eine Projektarbeit im Rahmen einer Umschulung etwas zu umfangreich geraten ist, entsprechend der Mehraufwand an Arbeitszeit. Der Lerneffekt aus dieser Arbeit war für mich sehr groß und übertrifft meine Erwartungen. Schwierigkeiten gab es hauptsächlich mit dem rndc.key des DNS-Servers im Hinblick auf den Imitationsserver, die aber durch eine Weiterleitung an die Clients gelöst wurde.

### 7.2 AUSBLICK

Nach meiner Einschätzung kann das Pilotprojekt in dieser Form jederzeit übernommen und in den Produktiveinsatz gebracht werden.

## 8. KUNDENDOKUMENTATION

Auf der Kundendokumentation stellt die Funktionalität für die Nutzung von einem Test Client auf der Testumgebung und auf der Webanwendung.

Siehe [Kundendokumentation Anlage](#) für weitere Details.

## 9. ADMIN-DOKUMENTATION

Auf der Admin-Dokumentation stellt die Nutzung von der System für Einfügen, Lesen und Löschen von einem Server auf der Testumgebung als einen einzigen Server oder auf eine Load-Balancing.

Siehe [Admin-Dokumentation Anlage](#) für weitere Details.



## GLOSSAR

### ACL (ACCESS CONTROL LIST)

Eine **Access Control List (ACL)**; deutsch **Zugriffssteuerungsliste, ZSL<sup>[1]</sup>**), ist eine **Software**-Technik, mit der **Betriebssysteme** und **Anwendungsprogramme** Zugriffe auf Daten und Funktionen eingrenzen können. Eine ACL legt fest, in welchem Umfang einzelne **Benutzer** und System**prozesse** Zugriff auf bestimmte Objekte (wie Dienste, **Dateien**, **Registrier**-Einträge etc.) haben.

Im Unterschied zu einfachen **Zugriffsrechten** sind ACLs feiner einstellbar. So können etwa bei Linux für eine Datei für mehrere Benutzer und Gruppen unterschiedliche Rechte vergeben werden, während reguläre Zugriffsrechte nur die Rechtevergabe für einen Benutzer, eine Gruppe und den „Rest der Welt“ zulassen. (Wikipedia, 2016)

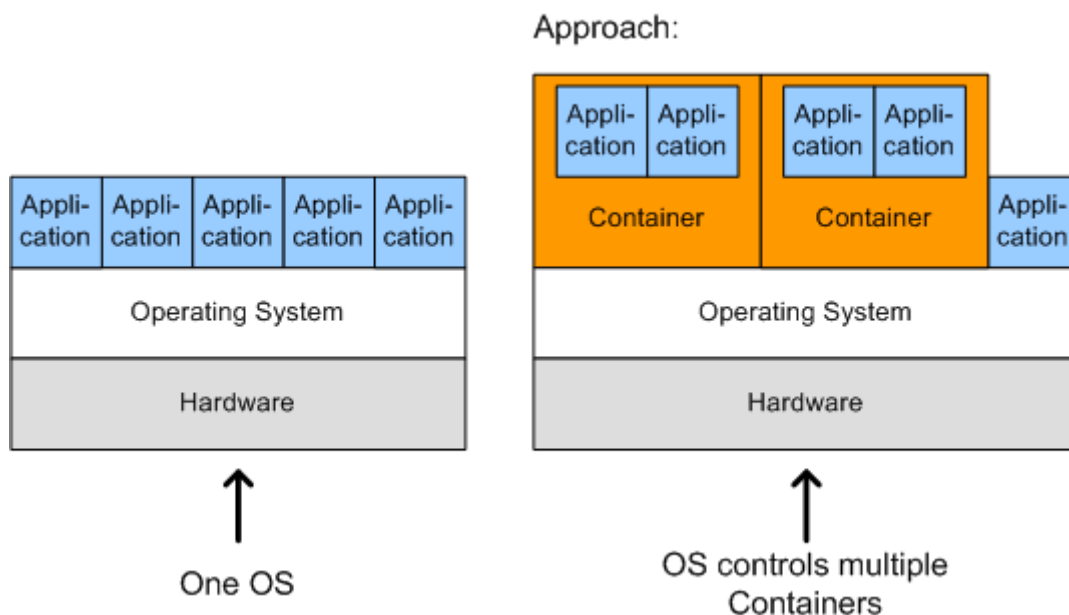
### ATOM EDITOR

<https://atom.io/>

Atom ist ein **Open Source-Texteditor** auf Basis von **Electron**. Electron besteht aus dem Webbrowser **Chromium** und dem JavaScript-Framework **Node.js** und erlaubt, beliebige Anwendungen aus JavaScript, HTML und CSS zu erstellen. Atom integriert einen **Paketmanager** namens **apm** und **Git**-Versionsverwaltung. Aufgrund der Rendering-Engine als Unterbau bietet Atom **Syntaxhervorhebung** für viele Programmiersprachen und erlaubt den Anwendern, das Programm beliebig mit Plugins und Themes zu erweitern (Wikipedia, 2016)

### CONTAINER

Betriebssystemvirtualisierung mittels OS-Container



Ansatz von 'OS-Containern' zur Schaffung virtueller Betriebsumgebungen.

Bei Virtualisierung auf Betriebssystemebene wird anderen **Computerprogrammen** eine komplette **Laufzeitumgebung** virtuell innerhalb eines geschlossenen Containers (unter BSD: **Jail**, unter Solaris: **Zone**) zur Verfügung gestellt. Es wird kein weiteres Betriebssystem gestartet; deshalb ist es unmöglich, verschiedene Betriebssysteme (oder auch nur verschiedene Versionen desselben Betriebssystems) gleichzeitig als Gastbetriebssysteme zu betreiben - der Hauptnachteil dieses Konzepts. In Containern können auch keine Treiber geladen werden. Der Vorteil dieses Konzepts besteht wegen des Verzichts auf einen Hypervisor darin,

besonders effizient mit den Ressourcen umzugehen (insbesondere hinsichtlich der Prozessor-Last und des Haupt- und Massenspeicherbedarfs), weshalb man auch von „leichtgewichtiger“ Virtualisierung spricht. Bei der Betriebssystemvirtualisierung läuft eben immer nur ein Host-Kernel, wobei [UML](#) eine gewisse Sonderrolle zukommt, da dort ein spezieller User-Mode-Kernel unter der Kontrolle des Host-Kernels läuft.

(Wikipedia, 2016)

## COREOS

<https://coreos.com/>

CoreOS ist ein Open-Source leichte Betriebssystem auf dem Linux-Kernel basieren und für Infrastruktur zu geclusterten Installationen bietet, während auf der Automatisierung konzentriert, die einfache Bereitstellung von Anwendungen, Sicherheit, Zuverlässigkeit und Skalierbarkeit. Als Betriebssystem liefert CoreOS nur die minimale Funktionalität die Bereitstellung von Anwendungen innerhalb Software-Container zusammen mit eingebauten Mechanismen für Service Discovery und Konfigurationsfreigabe erforderlich für. CoreOS teilt die Fundamente mit Chrome OS und Chromium OS, mit den Mitteln ihrer gemeinsamen Softwareentwicklung Kit (SDK) als Basis, während das Hinzufügen neuer Funktionen und Customizing es Hardware-in-Servern verwendet zu unterstützen. Ab Januar 2015 wird CoreOS aktiv mit den wichtigsten Funktionen zur Verfügung als stabile Version entwickelt, vor allem von Alex Polvi, Brandon Philips und Michael Marineau. (Wikipedia, 2016)

## DEBIAN

**Debian** ([englisch](#) ['debɪən]) ist ein gemeinschaftlich entwickeltes [freies Betriebssystem](#). **Debian GNU/Linux** basiert auf den grundlegenden Systemwerkzeugen des [GNU](#)-Projektes sowie dem [Linux-Kernel](#). Die aktuelle Version ist Debian 8 „Jessie“. Debian enthält eine große Auswahl an Anwendungsprogrammen und Werkzeugen; derzeit sind es über 43.000 [Programmpakete](#).

Debian wurde im August 1993 von [Ian Murdock](#) ins Leben gerufen und wird seitdem aktiv weiterentwickelt. Heute hat das Projekt über 1.000 offizielle Entwickler. Es ist eine der ältesten, einflussreichsten und am weitesten verbreiteten [GNU/Linux-Distributionen](#). Viele weitere Distributionen benutzen Debian als Grundlage. Das heute bekannteste Debian-GNU/Linux-[Derivat](#) ist [Ubuntu](#). (Wikipedia, 2016)

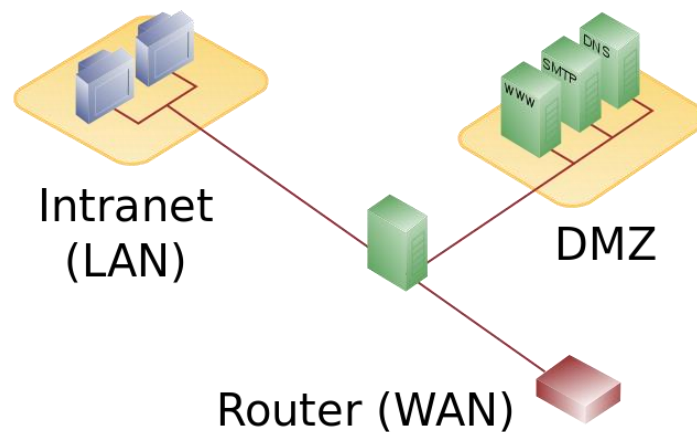
## DHCP (DYNAMIC HOST CONFIGURATION PROTOCOL)

Das DHCP ist ein [Kommunikationsprotokoll](#) in der [Computertechnik](#). Es ermöglicht die Zuweisung der Netzwerkkonfiguration an [Clients](#) durch einen [Server](#).

DHCP wurde im [RFC 2131](#) definiert und bekam von der [Internet Assigned Numbers Authority](#) die [UDP-Ports](#) 67 und 68 zugewiesen. (Wikipedia, 2016)

## DMZ (DEMILITARIZED ZONE)

Eine Demilitarized Zone (DMZ, auch *ent- oder demilitarisierte Zone*) bezeichnet ein [Computernetz](#) mit sicherheitstechnisch kontrollierten Zugriffsmöglichkeiten auf die daran angeschlossenen [Server](#).



Die in der DMZ aufgestellten Systeme werden durch eine oder mehrere [Firewalls](#) gegen andere Netze (z. B. [Internet](#), [LAN](#)) abgeschirmt. Durch diese Trennung kann der Zugriff auf öffentlich erreichbare Dienste ([Bastion Hosts](#) mit z. B. [E-Mail](#), [WWW](#) o. ä.) gestattet und gleichzeitig das interne Netz (LAN) vor unberechtigten Zugriffen von außen geschützt werden.

Der Sinn besteht darin, auf möglichst sicherer Basis Dienste des Rechnernetzes sowohl dem [WAN](#) ([Internet](#)) als auch dem LAN ([Intranet](#)) zur Verfügung zu stellen.

Ihre Schutzwirkung entfaltet eine DMZ durch die [Isolation](#) eines Systems gegenüber zwei oder mehr Netzen. (Wikipedia, 2016)

## DIG

Ist ein Befehl in verschiedenen Betriebssystemen, mit dem die Server des weltweiten [Domain Name Systems](#) (DNS) abgefragt werden können.

Der Name ist die Abkürzung von „domain information groper“ (wörtlich aus dem Englischen übersetzt: „Abtaster für Domain-Informationen“, aber auch ein [Wortspiel](#) mit *to dig*, „ausgraben“).

*dig* wird als modernere Alternative zu [nslookup](#) angesehen. Es verzichtet auf den interaktiven Betriebsmodus von *nslookup*, bietet aber, insbesondere als Bestandteil der [quelloffenen BIND v9.x](#)-Distribution des [ISC](#) (BIND-tools), umfassendere Abfragemöglichkeiten für praktisch alle in Frage kommenden DNS-Informationen. Damit ist es ein mächtiges Werkzeug für Test und Fehlersuche an DNS-Installationen, kann aber auch flexibel als [Kommandozeilen](#)-Tool und in [Shell-Skripts](#) verwendet werden. Ein besonderer Vorzug von (ISC-BIND-) *dig* ist dabei die weitestgehend identische Syntax seiner [Befehlszeilenparameter](#) beim Einsatz unter allen unterstützten Betriebssystemen. (Wikipedia, 2016)

## DNS (DOMAIN NAME SERVER)

Das DNS ist ein weltweit auf tausenden von [Servern](#) verteilter hierarchischer [Verzeichnisdienst](#), der den [Namensraum](#) des Internets verwaltet. Dieser Namensraum ist in so genannte [Zonen](#) unterteilt, für die jeweils unabhängige Administratoren zuständig sind. Für lokale Anforderungen – etwa innerhalb eines Firmennetzes – ist es auch möglich, ein vom Internet unabhängiges DNS zu betreiben.

Hauptsächlich wird das DNS zur Umsetzung von Domainnamen in IP-Adressen („*forward lookup*“) benutzt. Dies ist vergleichbar mit einem Telefonbuch, das die Namen der Teilnehmer in ihre Telefonnummer auflöst. Das DNS bietet somit eine Vereinfachung, weil Menschen sich Namen weitaus besser merken können als Zahlenkolonnen. So kann man sich einen Domainnamen wie *example.org* in der Regel leichter merken als die dazugehörige IP-Adresse 192.0.32.10. Dieser Punkt gewinnt im Zuge der Einführung von [IPv6](#) noch an Bedeutung, denn dann werden einem Namen jeweils IPv4- und IPv6-Adressen zugeordnet. So löst sich

beispielsweise der Name *www.kame.net* in die IPv4-Adresse 203.178.141.194 und die IPv6-Adresse 2001:200:0:8002:203:47ff:fea5:3085 auf.

Ein weiterer Vorteil ist, dass IP-Adressen – etwa von Web-Servern – relativ risikolos geändert werden können. Da Internetteilnehmer nur den (unveränderten) DNS-Namen ansprechen, bleiben ihnen Änderungen der untergeordneten IP-Ebene weitestgehend verborgen. Da einem Namen auch mehrere IP-Adressen zugeordnet werden können, kann sogar eine einfache [Lastverteilung per DNS](#) (*Load Balancing*) realisiert werden.

Mit dem DNS ist auch eine umgekehrte Auflösung von IP-Adressen in Namen (*reverse lookup*) möglich. In Analogie zum Telefonbuch entspricht dies einer Suche nach dem Namen eines Teilnehmers zu einer bekannten Rufnummer, was innerhalb der Telekommunikationsbranche unter dem Namen [Inverssuche](#) bekannt ist.

Das DNS wurde 1983 von [Paul Mockapetris](#) entworfen und in [RFC 882](#) und [RFC 883](#) (RFC = [Request for Comments](#)) beschrieben. Beide wurden inzwischen von [RFC 1034](#) und [RFC 1035](#) abgelöst und durch zahlreiche weitere Standards ergänzt. Ursprüngliche Aufgabe war es, die lokalen [hosts](#)-Dateien abzulösen, die bis dahin für die [Namensauflösung](#) zuständig waren und die der enorm zunehmenden Zahl von Neueinträgen nicht mehr gewachsen waren. Aufgrund der erwiesenermaßen hohen Zuverlässigkeit und Flexibilität wurden nach und nach weitere Datenbestände in das DNS integriert und so den Internetnutzern zur Verfügung gestellt (siehe unten: [Erweiterung des DNS](#)).

DNS zeichnet sich aus durch:

- dezentrale Verwaltung,
- hierarchische Strukturierung des Namensraums in Baumform,
- Eindeutigkeit der Namen,
- Erweiterbarkeit.

(Wikipedia, 2016)

## DNSDOMAINNAME

druckt der Domain-Teil des FQDN (Fully Qualified Domain Name). Der komplette FQDN des Systems wird mit dem Hostnamen zurückgegeben. (Linux-Command.org, 2016)

## DOCKER

Docker ist eine [Open-Source-Software](#), die dazu verwendet werden kann, Anwendungen mithilfe von Betriebssystemvirtualisierung in [Containern](#) zu isolieren. Dies vereinfacht einerseits die Bereitstellung von Anwendungen, weil sich Container, die alle nötigen Pakete enthalten, leicht als Dateien transportieren und installieren lassen. Andererseits gewährleisten Container die Trennung der auf einem Rechner genutzten Ressourcen, sodass ein Container keinen Zugriff auf Ressourcen anderer Container hat. (Wikipedia, 2016)

## FIREWALL

Eine Firewall (von [englisch](#) *firewall* ['faɪəwɔ:l], ‚Brandwand‘ oder ‚Brandmauer‘) ist ein Sicherungssystem, das ein [Rechnernetz](#) oder einen einzelnen Computer vor unerwünschten Netzwerkzugriffen schützt und ist weiter gefasst auch [ein Teilaspekt](#) eines [Sicherheitskonzepts](#). (Wikipedia, 2016)

## GITHUB

<https://github.com/>

GitHub ist ein webbasierter [Online-Dienst](#), der [Software](#)-Entwicklungsprojekte auf seinen Servern bereitstellt ([Filehosting](#)). Namensgebend war das [Versionsverwaltungssystem Git](#). (Wikipedia, 2016)

## HOSTNAME

Der **Hostname** (auch **Sitenname**) ist die eindeutige Bezeichnung eines [Rechners](#) in einem [Netzwerk](#). Er wird vorwiegend beim elektronischen Datenaustausch (z. B. [E-Mail](#), [Usenet](#), [FTP](#)) benutzt, um den Kommunikationspartner in einem von Menschen les- und merkbaren Format anzugeben. Die Umsetzung des Hostnamens in eine maschinenlesbare Adresse erfolgt im Internet heute vorwiegend über das [Domain Name System \(DNS\)](#), historisch über die [hosts](#)-Datei. In lokalen Netzen erfolgt die Umsetzung teils per [DNS](#), teils noch mit [NIS](#) und anderen proprietären Protokollen. Welche Namen technisch zulässig sind, regelt das im jeweiligen Fall verwendete Protokoll zur Namensauflösung – im Internet also das [Domain Name System](#). (Wikipedia, 2016)

## IDE (INTEGRATED DEVELOPMENT ENVIRONMENT)

Eine integrierte Entwicklungsumgebung (Abkürzung IDE, von [englisch](#) *integrated development environment*, auch (als Teilaspekte) *integrated design environment* oder *integrated debugging environment*) ist eine Sammlung von [Anwendungsprogrammen](#), mit denen die Aufgaben der [Softwareentwicklung](#) (SWE) möglichst ohne [Medienbrüche](#) bearbeitet werden können.

Integrierte Entwicklungsumgebungen für den SWE-Teilbereich [Programmierung](#) verfügen in der Regel über folgende Komponenten:

- [Texteditor](#)
- [Compiler](#) bzw. [Interpreter](#)
- [Linker](#)
- [Debugger](#)
- [Quelltextformatierungsfunktion](#)

(Wikipedia, 2016)

## GUI (GRAFISCHE BENUTZEROBERFLÄCHE)

Grafische Benutzeroberfläche oder auch grafische Benutzerschnittstelle (Abk. GUI von [englisch](#) *graphical user interface*) bezeichnet eine Form von [Benutzerschnittstelle](#) eines [Computers](#). Sie hat die Aufgabe, [Anwendungssoftware](#) auf einem Rechner mittels grafischer [Symbole](#), [Steuerelemente](#) oder auch [Widgets](#) genannt, bedienbar zu machen. Dies geschieht bei [Computern](#) meistens mittels einer [Maus](#) als Steuergerät, mit der die grafischen Elemente bedient oder ausgewählt werden, bei [Smartphones](#), [Tablets](#) und [Kiosksystemen](#) in der Regel durch Berührung eines [Sensorbildschirms](#). (Wikipedia, 2016)

## IP-ADRESSE (INTERNET PROTOCOL ADDRESS)

Eine IP-Adresse ist eine [Adresse](#) in [Computernetzen](#), die – wie das [Internet](#) – auf dem [Internetprotokoll](#) (IP) basiert. Sie wird Geräten zugewiesen, die an das Netz angebunden sind, und macht die Geräte so adressierbar und damit erreichbar. Die IP-Adresse kann einen einzelnen Empfänger oder eine Gruppe von Empfängern bezeichnen ([Multicast](#), [Broadcast](#)). Umgekehrt können einem Computer mehrere IP-Adressen zugeordnet sein.

Die IP-Adresse wird verwendet, um Daten von ihrem Absender zum vorgesehenen Empfänger transportieren zu können. Ähnlich der [Postanschrift](#) auf einem Briefumschlag werden [Datenpakete](#) mit einer IP-Adresse versehen, die den Empfänger eindeutig identifiziert. Aufgrund dieser Adresse können die „Poststellen“, die [Router](#), entscheiden, in welche Richtung das Paket weitertransportiert werden soll. Im Gegensatz zu Postadressen sind IP-Adressen nicht an einen bestimmten Ort gebunden.

Die bekannteste Notation der heute geläufigen *IPv4-Adressen* besteht aus vier Zahlen, die Werte von 0 bis 255 annehmen können und mit einem Punkt getrennt werden, beispielsweise 192.0.2.42. Technisch gesehen ist die Adresse eine 32-stellige ([IPv4](#)) oder 128-stellige ([IPv6](#)) [Binärzahl](#).

## IPCONFIG

Ist ein Befehl des Betriebssystems [Microsoft Windows](#) (ab den netzwerkfähigen Versionen Windows NT und Windows 2000), der im [Befehlseingabemodus](#) (Startmenü: Ausführen "cmd") die Hardwareadressen bzw. die IP-Adressen der im lokalen Netzwerk verwendeten Geräte anzeigt. (Wikipedia, 2016)

## ISO-ABBILD

ISO-Abbild (engl. *ISO image*) ist die Bezeichnung für eine [Computer-Datei](#), die ein [Speicherabbild](#) des [Dateisystems](#) einer [CD](#) oder [DVD](#) enthält, die im Format [ISO 9660](#) strukturiert ist.

Das [Dateisystem](#) wird beim Erzeugen eines ISO-Abbildes unverändert kopiert, dadurch bleiben Berechtigungen und andere [Metadaten](#) unverändert erhalten.

Die Norm [ISO 9660](#) definiert ein Dateisystem für CDs bzw. DVDs. (Wikipedia, 2016)

## LINUX

Als Linux oder GNU/Linux (*siehe* [GNU/Linux-Namensstreit](#)) bezeichnet man in der Regel [freie, unix-ähnliche Mehrbenutzer-Betriebssysteme](#), die auf dem [Linux-Kernel](#) und wesentlich auf [GNU](#)-Software basieren. Die weite, auch kommerzielle Verbreitung wurde ab 1992 durch die Lizenzierung des Linux-Kernels unter der freien Lizenz [GPL](#) ermöglicht. Einer der Initiatoren von Linux war der finnische Programmierer [Linus Torvalds](#). Er nimmt bis heute eine koordinierende Rolle bei der Weiterentwicklung des Linux-Kernels ein und wird auch als [Benevolent Dictator for Life](#) (deutsch *wohlwollender Diktator auf Lebenszeit*) bezeichnet. (Wikipedia, 2016)

## LINUX MINT

Linux Mint ist eine [Linux-Distribution](#) für [PC \(32-bit & 64-bit\)](#) in zwei parallel verfügbaren Ausgaben. Die Hauptausgabe, einfach *Linux Mint* genannt, basiert auf der Linux-Distribution [Ubuntu](#) und ist in zwei Varianten verfügbar, nämlich als [Cinnamon](#)- und als [MATE](#)-Edition. Die zweite (äußerlich sehr ähnliche) Ausgabe beruht auf [Debian](#) und wird zur Unterscheidung *Linux Mint Debian Edition (LMDE)* genannt. [Maintainer](#) für Linux Mint und LMDE ist Clément Lefebvre. Linux Mint wird auf der Website [Distrowatch](#) seit 2011 als die am häufigsten [aufgerufene Linux-Distribution](#) geführt. (Wikipedia, 2016)

## LOGDATEI

Eine Logdatei (auch *Ereignisprotokolldatei*; [englisch](#) *log file*) enthält das automatisch geführte [Protokoll](#) aller oder bestimmter Aktionen von [Prozessen](#) auf einem Computersystem. Die korrekte Bezeichnung dafür ist deshalb Protokolldatei.

Wichtige Anwendungen finden sich vor allem bei der Prozesskontrolle und Automatisierung. Prinzipiell werden alle Aktionen mitgeschrieben, die für eine spätere Untersuchung ([Audit](#)) erforderlich sind oder sein könnten. Der [Flugschreiber](#) in Flugzeugen ist ein Beispiel für kontinuierliche Protokollierung, die jedoch selten ausgewertet wird, zum Beispiel nach einem Unfall. (Wikipedia, 2016)

## MAC-ADRESSE (MEDIA-ACCESS-CONTROL)

Die MAC-Adresse ([Media-Access-Control](#)-Adresse) ist die Hardware-Adresse jedes einzelnen [Netzwerkadapters](#), die als eindeutiger [Identifikator](#) des Geräts in einem [Rechnernetz](#) dient. Bei [Apple](#) wird sie auch *Ethernet-ID*, *Airport-ID* oder *Wi-Fi-Adresse* genannt, bei [Microsoft](#) *Physikalische Adresse*. (Wikipedia, 2016)

## MAC-ADRESSE (MEDIA-ACCESS-CONTROL)

Die MAC-Adresse ([Media-Access-Control](#)-Adresse) ist die Hardware-Adresse jedes einzelnen [Netzwerkadapters](#), die als eindeutiger [Identifikator](#) des Geräts in einem [Rechnernetz](#) dient. Bei [Apple](#) wird sie auch *Ethernet-ID*, *Airport-ID* oder *Wi-Fi-Adresse* genannt, bei [Microsoft](#) *Physikalische Adresse*. (Wikipedia, 2016)

## NMAP

Ist ein Werkzeug zum Scannen und Auswerten von [Hosts](#) in einem Computernetzwerk und fällt somit in die Kategorie der [Portscanner](#). Der Name steht für *Network Mapper*. (Wikipedia, 2016)

## NSLOOKUP

Der Befehl nslookup kann unter [Mac OS X](#), [Windows](#) und [Unix](#) verwendet werden, um [IP-Adressen](#) oder [Domains](#) eines bestimmten Computers mittels [DNS](#) herauszufinden. Der Name des Befehls bedeutet „Name Server look up“, was so viel heißt wie „beim Namens-Server nachschauen“. Als modernere Alternative zu nslookup hat sich zunehmend der Befehl [dig](#) etabliert, der unter Windows als [Cygwin](#)-Port verfügbar ist. (Wikipedia, 2016)

## PORTWERTEILUNG

Eine Portweiterleitung (englisch *port forwarding*) ist die Weiterleitung einer Verbindung, die über ein [Rechnernetz](#) auf einen bestimmten [Port](#) eingeht, zu einem anderen [Computer](#). Da der entsprechende [Netzwerkdienst](#) nicht von dem weiterleitenden Computer selbst geleistet wird, benutzt man hier auch irreführend den Begriff *virtueller Server*.

Die eingehenden [Datenpakete](#) werden hierbei per [Destination NAT](#) und die ausgehenden Pakete per [Source NAT](#) maskiert, um sie an den anderen Rechner weiterzuleiten bzw. den Anschein zu erwecken, die ausgehenden Pakete kämen von dem Computer, der die Portweiterleitung betreibt.

Port Forwarding wird oft dazu benutzt, [FTP](#), [Web-Server](#) oder andere [Server](#)-basierende Anwendungen hinter einem [NAT](#)-Gateway zu betreiben. (Wikipedia, 2016)

## PROXY

Ein Proxy (von [englisch](#) *proxy representative* „Stellvertreter“, von [lateinisch](#) *proximus* „der Nächste“) ist eine [Kommunikationsschnittstelle](#) in einem [Netzwerk](#). Er arbeitet als Vermittler, der auf der einen Seite Anfragen entgegennimmt, um dann über seine eigene Adresse eine Verbindung zur anderen Seite herzustellen.

Wird der [Proxy als Netzwerkkomponente](#) eingesetzt, bleibt einerseits die tatsächliche Adresse eines Kommunikationspartners dem jeweils anderen Kommunikationspartner verborgen, was eine gewisse Anonymität schafft. Als (mögliches) Verbindungsglied zwischen unterschiedlichen Netzwerken kann er andererseits eine Verbindung zwischen Kommunikationspartnern selbst dann realisieren, wenn deren Adressen zueinander inkompatibel sind und eine direkte Verbindung nicht möglich ist. (Wikipedia, 2016)

## PLUGINS

Ist ein optionales [Software-Modul](#), das eine bestehende [Software](#) erweitert bzw. verändert. Der Begriff wird teilweise auch als Synonym zu „Add-on“ benutzt. Plug-ins werden meist vom [Benutzer](#) installiert und dann von der entsprechenden Hauptanwendung während der Laufzeit eingebunden. Plug-ins können nicht ohne die Hauptanwendung ausgeführt werden. (Linux-Command.org, 2016)

## SQUID



Ist ein [freier Proxyserver](#) und [Web-Cache](#), der unter der [GNU General Public Licence](#) steht. Er zeichnet sich vor allem durch seine gute [Skalierbarkeit](#) aus. Squid unterstützt die [Netzwerkprotokolle HTTP/HTTPS, FTP](#) über HTTP und [Gopher](#). (Wikipedia, 2016)

## UBUNTU

Ubuntu (auch: Ubuntu Linux) ist eine kostenlose [Linux-Distribution](#), die auf [Debian](#) basiert. Der Name Ubuntu bedeutet auf [Zulu](#) etwa „Menschlichkeit“ und bezeichnet eine [afrikanische Philosophie](#). Die Entwickler verfolgen mit Ubuntu das Ziel, ein einfach zu installierendes und leicht zu bedienendes [Betriebssystem](#) mit aufeinander abgestimmter Software zu schaffen. Dies soll unter anderem dadurch erreicht werden, dass für jede Aufgabe genau ein Programm zur Verfügung gestellt wird. Das Projekt wird vom Software-Hersteller [Canonical Ltd.](#) gesponsert, der vom [südafrikanischen](#) Unternehmer [Mark Shuttleworth](#) gegründet wurde. (Wikipedia, 2016)

## UBUNTU MATE

Die Variante Ubuntu MATE verwendet die Desktop-Umgebung [MATE](#) an Stelle von Unity. Ab Version 14.10 ist Ubuntu MATE eine von Canonical offiziell anerkannte Variante. Das enthaltene Tool *MATE Tweak* bietet vorkonfigurierte Panel/Dock-Varianten mit unterschiedlichen Startmenüs zur Auswahl an. Systemanwendungen wie Dateimanager, Texteditor und Dokumentenbetrachter stehen als eigene, von GNOME abgeleitete, Anwendungen zur Verfügung, deren Namen in MATE hauptsächlich mit spanischen Begriffen wie *caja*, *pluma* und *atril* belegt sind. (Wikipedia, 2016)

## VIRTUAL BOX

Ist eine [Virtualisierungssoftware](#) des [US-amerikanischen](#) Unternehmens [Oracle](#). Sie kann auf den [Betriebssystemen FreeBSD, Linux, OS/2, OS X, Solaris](#) und [Windows](#) als Wirtssystem auf 32-Bit- oder 64-Bit-[x86](#)-Systemen eingesetzt werden. Als Gastsysteme werden eine Vielzahl an x86- und AMD64-Betriebssystemen unterstützt. Diese stehen bei der Einrichtung einer neuen [virtuellen Maschine](#) zur Auswahl. (Wikipedia, 2016)

## TRANSPARENT-PROXY

Ein *Transparenter Proxy* besteht grundsätzlich aus zwei Komponenten. Zunächst werden am [Router](#) die gewünschten Ports der Protokolle abgegriffen (beispielsweise über [iptables](#) unter Einsatz eines Redirects) und dann an einen Proxy weitergeleitet. Für den Anwender ist die Verbindung über einen transparenten Proxy in der Benutzung nicht von einer direkten Verbindung über den Router zu unterscheiden. Das Vorhandensein eines transparenten Proxys bietet daher den Nutzen, dass eine Konfiguration der Proxyeinstellungen am einzelnen PC unterbleiben kann (siehe unter [Sichtbarkeiten](#)). (Wikipedia, 2016)

## VAGRANT

Ist eine [freie](#)- und [Open-Source Ruby](#)-Anwendung zum Erstellen und Verwalten von [virtuellen Maschinen](#). Vagrant ermöglicht einfaches [Deployment](#) insbesondere in der [Software](#)- und [Webentwicklung](#) und dient als [Wrapper](#) zwischen [Virtualisierungssoftware](#) wie [VirtualBox](#), [VMware](#) und [Hyper-V](#) und [Software-Configuration-Management](#)-Anwendungen beziehungsweise Systemkonfigurationswerkzeugen wie [Chef](#), [Saltstack](#) und [Puppet](#). (Wikipedia, 2016)

## WEBALIZER

Ist eine [Web Analytics](#) Software, die mit [Logdateianalyse](#) arbeitet und unter der [GPL](#) steht. Angeregt wurde das Projekt im Jahre 1997 durch Bradford L. Barrett. Die vom Webalizer generierten Statistiken enthalten üblicherweise folgende Informationen: Anfragen, Besuche, Verweise, Länder der Besucher und Menge der ausgelieferten Daten. Diese Statistiken können sowohl grafisch als auch textuell betrachtet werden und sind auf unterschiedlichen Zeitskalen (Stunden, Tage, Monate, Jahre) dargestellt. (Wikipedia, 2016)



## WHOIS

Ist ein [Protokoll](#), mit dem von einem verteilten [Datenbanksystem](#) Informationen zu [Internet-Domains](#) und [IP-Adressen](#) und deren Eigentümern abgefragt werden können.

*Whois*-Anfragen werden seit ihren Anfängen vor allem über die [Kommandozeile](#) durchgeführt. Da entsprechende Client-Software nicht für alle gängigen Betriebssysteme verfügbar war, setzten sich früh Web-basierte [Frontends](#) durch. Trotz späterer entsprechender Versionen erfreuen sich Web-Whois-Anbieter immer noch großer Beliebtheit, nicht zuletzt aus Gründen der Aktualität bei Domain-Lookups.

Aus Datenschutzgründen können Informationen über Eigentümer von .de Domains nicht mehr wie früher über das whois-Protokoll abgefragt werden. Dies geht nur noch über eine mit einem [Captcha](#) gesicherte Webwhois-Seite, erreichbar über die Homepage<sup>[1]</sup> des [DENIC](#).

Der Begriff „Whois“ wird auch für weitere, vergleichbare Abfragen verwendet, zum Beispiel Nutzerinformationen im [IRC](#) betreffend. (Wikipedia, 2016)

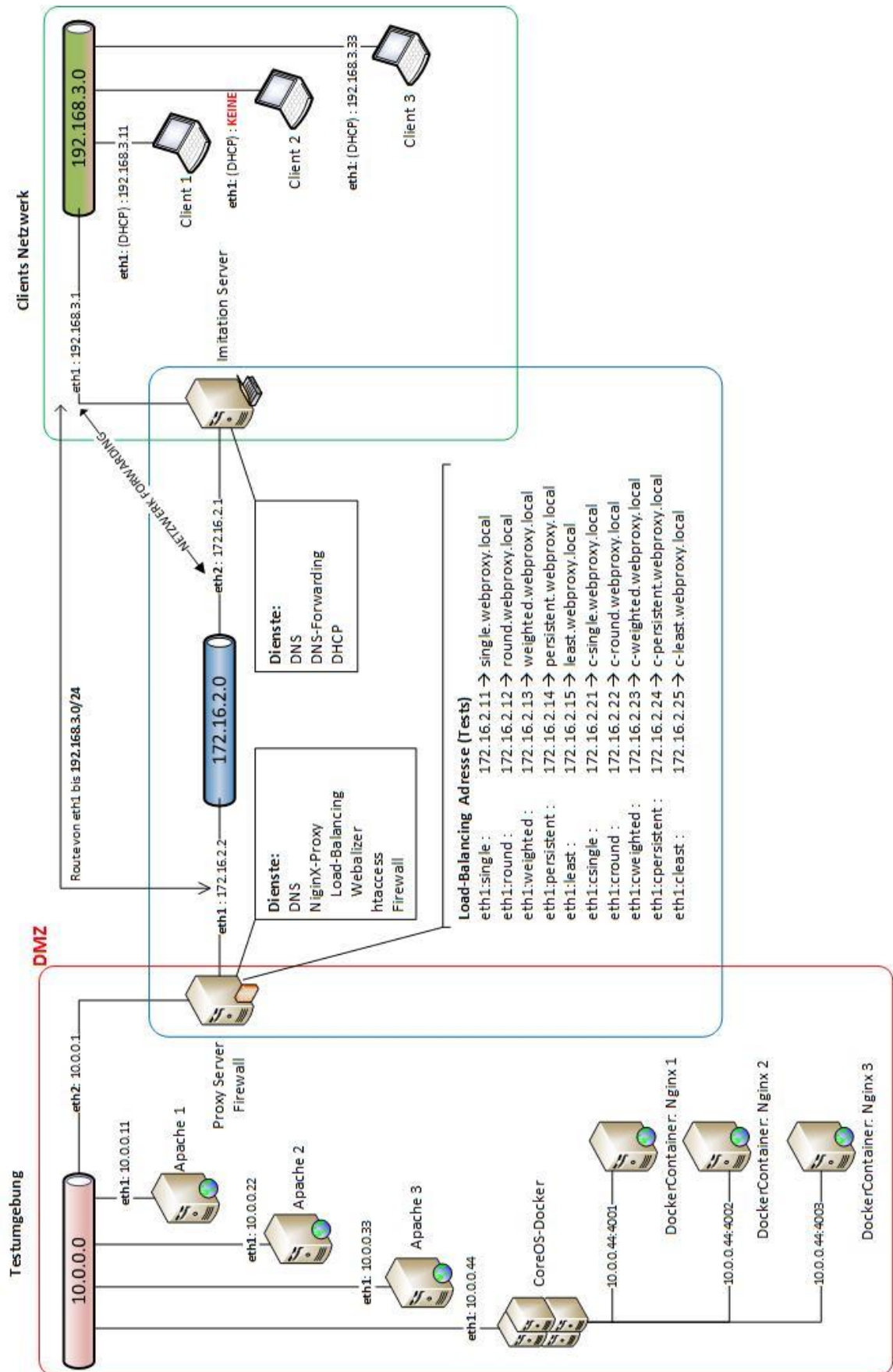
## WINDOWS 7

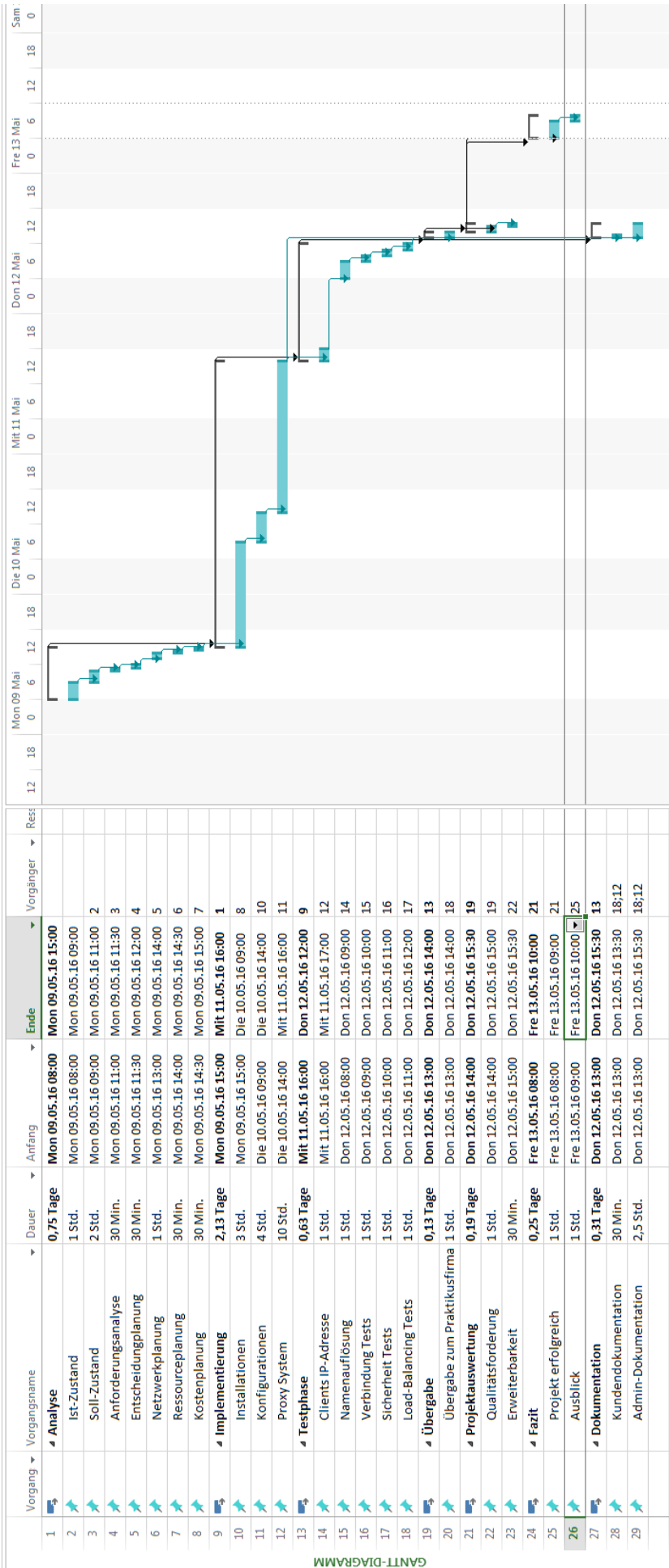
Microsoft Windows 7 ist ein [Betriebssystem](#) des [US-amerikanischen](#) Unternehmens [Microsoft](#). Es erschien am 22. Oktober 2009 und ist der Nachfolger von [Microsoft Windows Vista](#). Der weltweite [Marktanteil](#) von Windows 7 bei den Desktop-Betriebssystemen liegt bei 51,89 [Prozent](#) (Stand: April 2016). (Wikipedia, 2016)

## ANLAGES

### ANLAGE 1: NETZWERKPLANUNG

## Pilot Projekt Load-Balancing Proxy





## ANLAGE 3: TABELLE SOLL-IST ZEITABLEUFEN

Projektphase	Aufgabenbeschreibung	Soll- Stunden	Ist- Stunden	Diff
Analyse	Ist-Zustand	1 Stunden	1 Stunden	
	Soll-Zustand	2 Stunden	2 Stunden	
	Anforderungsanalyse	0,5 Stunden	0,5 Stunden	
	Entscheidungsplanung	0,5 Stunden	0,5 Stunden	
	Netzwerkplanung	1 Stunden	1 Stunden	
	Ressourceplanung	0,5 Stunden	0,5 Stunden	
	Kostenplanung	0,5 Stunden	0,5 Stunden	
Implementierung	Installationen	3 Stunden	2 Stunden	-1 Stunden
	Konfigurationen	4 Stunden	5 Stunden	+1 Stunden
	Proxy System	10 Stunden	18 Stunden	+8 Stunden
Testphase	Clients IP-Adresse	1 Stunden	0,5 Stunden	-0,5 Stunden
	Namenauflösung	1 Stunden	0,5 Stunden	-0,5 Stunden
	Verbindung Tests	1 Stunden	0,5 Stunden	-0,5 Stunden
	Sicherheit Tests	1 Stunden	0,5 Stunden	-0,5 Stunden
	Load-Balancing Tests	1 Stunden	0,5 Stunden	-0,5 Stunden
Übergabe	Übergabe zum Praktikumsfirma	1 Stunden	0,5 Stunden	-0,5 Stunden
Projektauswertung	Qualitätsforderung	1 Stunden	0,5 Stunden	-0,5 Stunden
	Erweiterbarkeit	0,5 Stunden	0,5 Stunden	
Fazit	Projekt erfolgreich	1 Stunden	0,5 Stunden	-0,5 Stunden
	Ausblick	1 Stunden	0,5 Stunden	-0,5 Stunde
Dokumentation	Kundendokumentation	0,5 Stunden	0,5 Stunden	
	Admin-Dokumentation	2,5 Stunden	1 Stunden	-1,5 Stunden
GESAMTZEIT		35 Stunden	37,5 Stunden	+2,5 Stunden

## KONFIGURATIONEN ANLAGEN

### KONFIGURATION ANLAGE 1: TEST CLIENT 1 KONFIGURATIONEN

---

#### CLIENT 1 NAME

```
web@cli1:~$ cat /etc/hostname
cli1
```

---

#### CLIENT 1 BEKANNTEN HOSTS

```
web@cli1:~$ cat /etc/hosts
127.0.0.1    localhost
127.0.1.1    cli1.name.local cli1
# The following lines are desirable for IPv6 capable hosts
::1         ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
```

---

#### CLIENT 1 DOMAINNAME

```
web@cli1:~$ cat /etc/resolvconf/resolv.conf.d/head
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by
resolvconf(8)

#      DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
domain name.local
```

---

#### CLIENT 1 NAMENAUFÖSUNG

```
web@cli1:~$ cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by
resolvconf(8)

#      DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
domain name.local
nameserver 192.168.3.1
search name.local
```

---

#### CLIENT 1 NETZWERKSCHNITTSTELLE

```
web@cli1:~$ cat /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

allow-hotplug enp0s3
```

```
iface enp0s3 inet dhcp
```

```
auto enp0s8
```

```
iface enp0s8 inet dhcp
```

## KONFIGURATION ANLAGE 2: TEST CLIENT 2 KONFIGURATIONEN

---

### CLIENT 2 NAME

```
web@cli2:~$ cat /etc/hostname  
cli2
```

---

### CLIENT 2 BEKANNTEN HOSTS

```
web@cli2:~$ cat /etc/hosts  
127.0.0.1    localhost  
127.0.1.1    cli2.name.local cli2  
# The following lines are desirable for IPv6 capable hosts  
::1         ip6-localhost ip6-loopback  
fe00::0     ip6-localnet  
ff00::0     ip6-mcastprefix  
ff02::1     ip6-allnodes  
ff02::2     ip6-allrouters
```

---

### CLIENT 2 DOMAINNAME

```
web@cli2:~$ cat /etc/resolvconf/resolv.conf.d/head  
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by  
resolvconf(8)  
#      DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN  
domain name.local
```

---

### CLIENT 2 NAMENAUFÖSUNG

```
web@cli2:~$ cat /etc/resolv.conf  
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by  
resolvconf(8)  
#      DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN  
domain name.local
```

---

### CLIENT 2 NETZWERKSCHNITTSTELLE

```
web@cli2:~$ cat /etc/network/interfaces  
# interfaces(5) file used by ifup(8) and ifdown(8)  
auto lo  
iface lo inet loopback  
  
allow-hotplug enp0s3
```

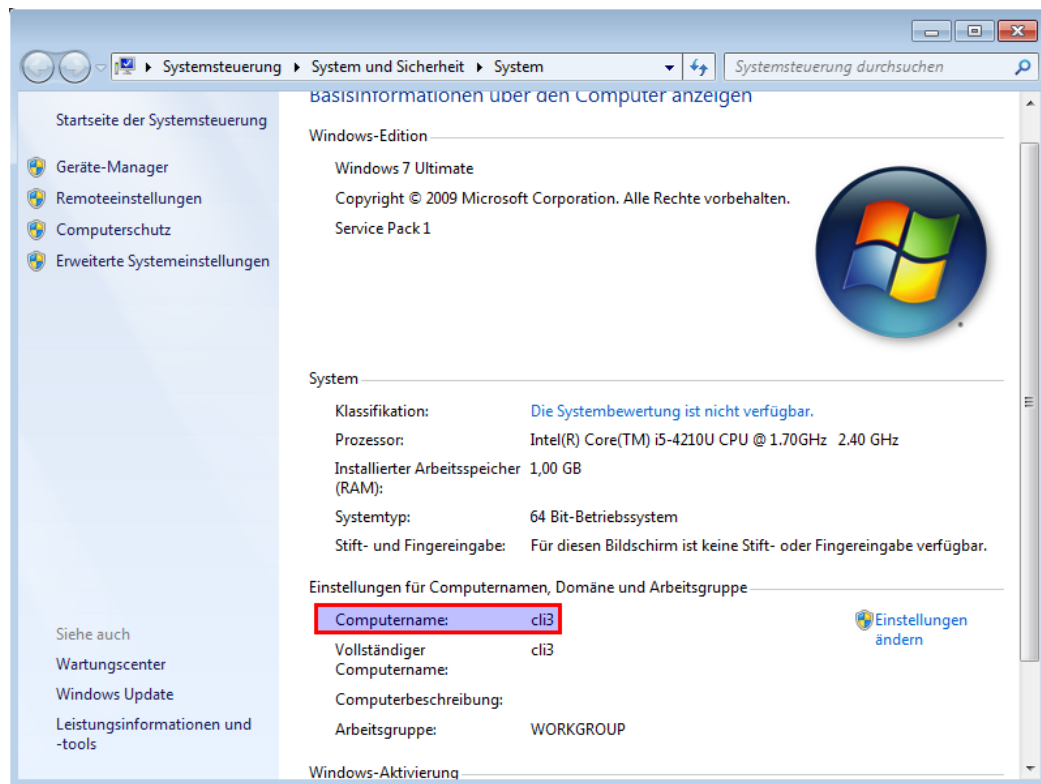
```
iface enp0s3 inet dhcp
```

```
auto enp0s8
```

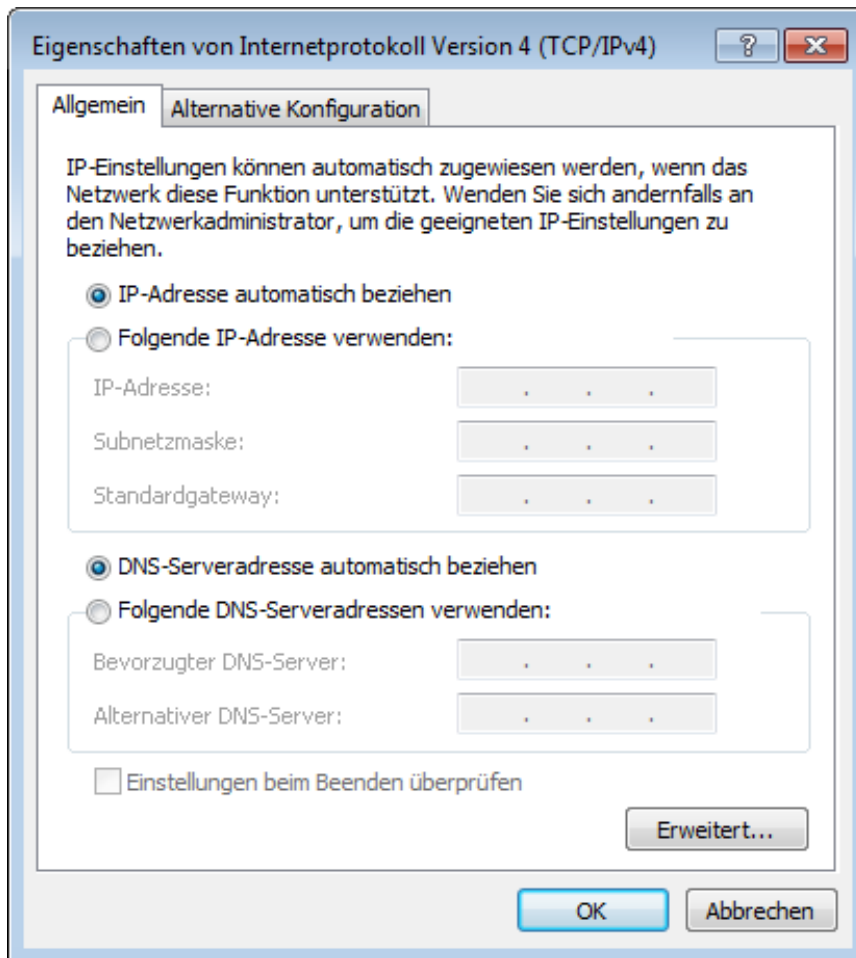
```
iface enp0s8 inet dhcp
```

## KONFIGURATION ANLAGE 3: TEST CLIENT 3 KONFIGURATIONEN

### CLIENT 3 NAME



### CLIENT 3 NETZWERKKONFIGURATIONEN



#### KONFIGURATION ANLAGE 4: TEST SERVER 1 KONFIGURATIONEN

---

##### TEST SERVER 1 NAME

```
root@apachel:~# cat /etc/hostname
apachel
```

---

##### TEST SERVER 1 BEKANNTEN HOSTS

```
root@apachel:~# cat /etc/hosts
127.0.0.1    localhost
127.0.1.1    apachel
10.0.0.11    apachel.webproxy.local apachel
# The following lines are desirable for IPv6 capable hosts
::1         localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
```

---

##### TEST SERVER 1 NETZWERKSCHNITTSTELLEN

```
root@apachel:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
```



```
# and how to activate them. For more information, see interfaces(5).
```

```
# The loopback network interface
```

```
auto lo
```

```
iface lo inet loopback
```

```
# The primary network interface
```

```
allow-hotplug eth0
```

```
iface eth0 inet dhcp
```

```
auto eth1
```

```
iface eth1 inet static
```

```
    address 10.0.0.11
```

```
    netmask 255.255.255.0
```

---

#### TEST SERVER 1 APACHE WEBSERVER ORDNER INHALT

```
root@apache1:~# cat /var/www/html/index.html
```

```
apache 1
```

#### KONFIGURATION ANLAGE 5: TEST SERVER 2 KONFIGURATIONEN

---

##### TEST SERVER 2 NAME

```
root@apache2:~# cat /etc/hostname
```

```
apache2
```

---

##### TEST SERVER 2 BEKANNTEN HOSTS

```
root@apache2:~# cat /etc/hosts
```

```
127.0.0.1    localhost
```

```
127.0.1.1    apache2
```

```
10.0.0.11    apache2.webproxy.local apache2
```

```
# The following lines are desirable for IPv6 capable hosts
```

```
::1          localhost ip6-localhost ip6-loopback
```

```
ff02::1     ip6-allnodes
```

```
ff02::2     ip6-allrouters
```

---

##### TEST SERVER 2 NETZWERKSCHNITTSTELLEN

```
root@apache2:~# cat /etc/network/interfaces
```

```
# This file describes the network interfaces available on your system
```

```
# and how to activate them. For more information, see interfaces(5).
```

```
# The loopback network interface
```

```
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
    address 10.0.0.22
    netmask 255.255.255.0
```

---

#### TEST SERVER 2 APACHE WEBSERVER ORDNER INHALT

```
root@apache2:~# cat /var/www/html/index.html
apache 2
```

#### KONFIGURATION ANLAGE 6: TEST SERVER 3 KONFIGURATIONEN

---

#### TEST SERVER 3 NAME

```
root@apache3:~# cat /etc/hostname
apache3
```

---

#### TEST SERVER 3 BEKANNTEN HOSTS

```
root@apache3:~# cat /etc/hosts
127.0.0.1    localhost
127.0.1.1    apache3
10.0.0.11    apache3.webproxy.local apache3
```

```
# The following lines are desirable for IPv6 capable hosts
::1         localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
```

---

#### TEST SERVER 3 NETZWERKSCHNITTSTELLEN

```
root@apache3:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback
```

```
# The primary network interface
allow-hotplug eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
    address 10.0.0.33
    netmask 255.255.255.0
```

---

#### TEST SERVER 3 APACHE WEBSERVER ORDNER INHALT

```
root@apache3:~# cat /var/www/html/index.html
apache 3
```

#### KONFIGURATION ANLAGE 7: IMITATION SERVER KONFIGURATIONEN

---

##### IMITATION SERVER NAME

```
root@dns:~# cat /etc/hostname
dns
```

---

##### IMITATION SERVER BEKANNTEN HOSTS

```
root@dns:~# cat /etc/hosts
127.0.0.1    localhost
127.0.1.1    dns
192.168.3.1  dns.name.local dns
```

```
# The following lines are desirable for IPv6 capable hosts
::1          localhost ip6-localhost ip6-loopback
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
```

---

##### IMITATION SERVER NETZWERKSCHNITTSTELLEN

```
root@dns:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet dhcp
```

```
auto eth1
iface eth1 inet static
    address 192.168.3.1
    netmask 255.255.255.0
```

```
auto eth2
iface eth2 inet static
    address 172.16.2.1
    netmask 255.255.255.0
```

---

## IMITATION SERVER PORTVERTEILUNG

```
root@dns:~# cat /etc/network/if-up.d/routingtabelle
#!/bin/bash
```

```
### FORWARDING
echo 1 > /proc/sys/net/ipv4/ip_forward
```

---

## IMITATION SERVER DNS KONFIGURATIONEN

---

### IMITATION SERVER RNDG.KEY

#### GENERIERUNG VON RNDG.KEY:

---

```
root@dns:~# rndc-confgen
# Start of rndc.conf
key "rndc-key" {
    algorithm hmac-md5;
    secret "????????????????????";
};

options {
    default-key "rndc-key";
    default-server 127.0.0.1;
    default-port 953;
};
# End of rndc.conf

# Use with the following in named.conf, adjusting the allow list as needed:
# key "rndc-key" {
#     algorithm hmac-md5;
#     secret "????????????????????";
# };
```

```
#
# controls {
#     inet 127.0.0.1 port 953
#         allow { 127.0.0.1; } keys { "rndc-key"; };
# };
# End of named.conf
```

#### EINSETZUNG VON GENERIERTE *RNDC.KEY* SCHLÜSSEL:

---

```
root@dns:~# cat /etc/bind/rndc.key
key "rndc-key" {
    algorithm hmac-md5;
    secret "????????????????????";
};
```

#### TESTEN KONFIGURATIONEN VON *RNDC.KEY*:

---

```
root@dns:~# rndc status
version: 9.9.5-3ubuntu0.8-Ubuntu <id:f9b8a50e>
CPUs found: 1
worker threads: 1
UDP listeners per interface: 1
number of zones: 101
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
recursive clients: 0/0/1000
tcp clients: 0/100
server is up and running
```

---

#### IMITATION SERVER *DB.NAME.LOCAL* DATEI

```
root@dns:~# cat /etc/bind/db.name.local
; BIND reverse data file for empty rfc1918 zone
;
; DO NOT EDIT THIS FILE - it is used for multiple zones.
; Instead, copy it, edit named.conf, and use that copy.
;
$TTL 86400
@      IN      SOA  dns.name.local. root.name.local. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
```

```

                2419200                ; Expire
                86400 )    ; Negative Cache TTL
;
@      IN      NS      dns.name.local.
@      IN      A       192.168.3.1
@      IN      AAAA    ::1
dns    IN      A       192.168.3.1
cli1   IN      A       192.168.3.11
cli2   IN      A       192.168.3.22
cli3   IN      A       192.168.3.33

```

---

#### IMITATION SERVER NAMED.CONF DATEI

```

root@dns:~# cat /etc/bind/named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in
/etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";

// Using rndc.key;
include "/etc/bind/rndc.key";
controls {
    inet 127.0.0.1 port 953 allow { localhost; } keys { "rndc-key"; };
};

```

---

#### IMITATION SERVER NAMED.CONF.LOCAL DATEI

```

root@dns:~# cat /etc/bind/named.conf.local
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

```

```
// IMITATION LOCAL DNS
zone "name.local" {
    type master;
    file "/etc/bind/db.name.local";
};
```

```
// FORWARD DNS TO PROXY
zone "webproxy.local" {
    type forward;
    forwarders {
        172.16.2.2;
    };
};
```

---

#### IMITATION SERVER NAMED.CONF.OPTIONS DATEI

```
root@dns:~# cat /etc/bind/named.conf.options
```

```
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    forwarders {
        172.16.2.2;
    };

    recursion yes;

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.  See https://www.isc.org/bind-keys
    //=====
    //dnssec-validation auto;
    dnssec-validation yes;
```

```
auth-nxdomain no;    # conform to RFC1035
listen-on-v6 { any; };
};
```

---

## IMITATION SERVER DHCP KONFIGURATIONEN

---

### IMITATION SERVER DHCPD.CONF DATEI

```
root@dns:~# cat /etc/dhcp/dhcpd.conf

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;


# option definitions common to all supported networks...
option domain-name "name.local";
option domain-name-servers dns.name.local;


default-lease-time 600;
max-lease-time 7200;


# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;


# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;


### CLIENTS NETWORK
deny unknown-clients;


shared-network intern {
    subnet 192.168.3.0 netmask 255.255.255.0 {
        range 192.168.3.10 192.168.3.100;
        option routers 192.168.3.1, 172.16.2.1;

        host client1 {
            hardware ethernet 08:00:27:36:41:21;
            fixed-address 192.168.3.11;
        }
    }
}
```



```

    }

# Client 2 werde es getestet al seine Client ohne Netzwerkzugriff
#   host client2 {
#       hardware ethernet 08:00:27:10:D9:08;
#       fixed-address 192.168.3.22;
#   }

    host client3 {
        hardware ethernet 08:00:27:A9:90:D7;
        fixed-address 192.168.3.33;
    }
}
}

```

## KONFIGURATION ANLAGE 8: CONTAINER SERVER KONFIGURATIONEN

### VAGRANT FILE

Der Änderung auf der Automatische generiert Konfigurationsdatei sind markiert.

```

# -*- mode: ruby -*-
# # vi: set ft=ruby :

require 'fileutils'

Vagrant.require_version ">= 1.6.0"

CLOUD_CONFIG_PATH = File.join(File.dirname(__FILE__), "user-data")
CONFIG = File.join(File.dirname(__FILE__), "config.rb")

# Defaults for config options defined in CONFIG
$num_instances = 1
$instance_name_prefix = "core"
$update_channel = "alpha"
$image_version = "current"
$enable_serial_logging = false
$share_home = false
$vm_gui = false
$vm_memory = 512
$vm_cpus = 1
$shared_folders = {}
$forwarded_ports = {}

```

```

# Attempt to apply the deprecated environment variable NUM_INSTANCES to
# $num_instances while allowing config.rb to override it
if ENV["NUM_INSTANCES"].to_i > 0 && ENV["NUM_INSTANCES"]
  $num_instances = ENV["NUM_INSTANCES"].to_i
end

if File.exist?(CONFIG)
  require CONFIG
end

# Use old vb_XXX config variables when set
def vm_gui
  $vb_gui.nil? ? $vm_gui : $vb_gui
end

def vm_memory
  $vb_memory.nil? ? $vm_memory : $vb_memory
end

def vm_cpus
  $vb_cpus.nil? ? $vm_cpus : $vb_cpus
end

Vagrant.configure("2") do |config|
  # always use Vagrants insecure key
  config.ssh.insert_key = false

  config.vm.box = "coreos-%s" % $update_channel
  if $image_version != "current"
    config.vm.box_version = $image_version
  end

  config.vm.box_url = "https://storage.googleapis.com/%s.release.core-
os.net/amd64-usr/%s/coreos_production_vagrant.json" % [$update_channel,
$image_version]

  ["vmware_fusion", "vmware_workstation"].each do |vmware|
    config.vm.provider vmware do |v, override|
      override.vm.box_url =
"https://storage.googleapis.com/%s.release.core-os.net/amd64-
usr/%s/coreos_production_vagrant_vmware_fusion.json" % [$update_channel,
$image_version]
    end
  end
end

```

```

    end
end

config.vm.provider :virtualbox do |v|
  # On VirtualBox, we don't have guest additions or a functional vboxsf
  # in CoreOS, so tell Vagrant that so it can be smarter.
  v.check_guest_additions = false
  v.functional_vboxsf      = false
end

# plugin conflict
if Vagrant.has_plugin?("vagrant-vbguest") then
  config.vbguest.auto_update = false
end

(1..$num_instances).each do |i|
  config.vm.define vm_name = "%s-%02d" % [$instance_name_prefix, i] do
|config|
    config.vm.hostname = vm_name

    if $enable_serial_logging
      logdir = File.join(File.dirname(__FILE__), "log")
      FileUtils.mkdir_p(logdir)

      serialFile = File.join(logdir, "%s-serial.txt" % vm_name)
      FileUtils.touch(serialFile)

      ["vmware_fusion", "vmware_workstation"].each do |vmware|
        config.vm.provider vmware do |v, override|
          v.vmx["serial0.present"] = "TRUE"
          v.vmx["serial0.fileType"] = "file"
          v.vmx["serial0.fileName"] = serialFile
          v.vmx["serial0.tryNoRxBios"] = "FALSE"
        end
      end
    end

    config.vm.provider :virtualbox do |vb, override|
      vb.customize ["modifyvm", :id, "--uart1", "0x3F8", "4"]
      vb.customize ["modifyvm", :id, "--uartmodel", serialFile]
    end
  end
end

```

```

    if $expose_docker_tcp
        config.vm.network "forwarded_port", guest: 2375, host:
($expose_docker_tcp + i - 1), auto_correct: true
    end

    $forwarded_ports.each do |guest, host|
        config.vm.network "forwarded_port", guest: guest, host: host,
auto_correct: true
    end

    ["vmware_fusion", "vmware_workstation"].each do |vmware|
        config.vm.provider vmware do |v|
            v.gui = vm_gui
            v.vmx['memsize'] = vm_memory
            v.vmx['numvcpus'] = vm_cpus
        end
    end

    config.vm.provider :virtualbox do |vb|
        vb.gui = vm_gui
        vb.memory = vm_memory
        vb.cpus = vm_cpus
    end

    ip = "10.0.0.44"
    config.vm.network :private_network, ip: ip,
    virtualbox____intnet: true

    # Uncomment below to enable NFS for sharing the host machine into the
coreos-vagrant VM.

    #config.vm.synced_folder ".", "/home/core/share", id: "core", :nfs =>
true, :mount_options => ['nolock,vers=3,udp']
    $shared_folders.each_with_index do |(host_folder, guest_folder),
index|
        config.vm.synced_folder host_folder.to_s, guest_folder.to_s, id:
"core-share%02d" % index, nfs: true, mount_options: ['nolock,vers=3,udp']
    end

    if $share_home
        config.vm.synced_folder ENV['HOME'], ENV['HOME'], id: "home", :nfs
=> true, :mount_options => ['nolock,vers=3,udp']
    end

```

```

end

    if File.exist?(CLOUD_CONFIG_PATH)
        config.vm.provision :file, :source => "#{CLOUD_CONFIG_PATH}",
:destination => "/tmp/vagrantfile-user-data"
        config.vm.provision :shell, :inline => "mv /tmp/vagrantfile-user-
data /var/lib/coreos-vagrant/", :privileged => true
    end
end
end
end

```

## KONFIGURATION ANLAGE 9: CONTAINER 1 KONFIGURATIONEN

### NGINX ORDNER INHALT

```

core@core-01 ~ $ cat nginx_servers/ng1/index.html
nginx_1 DOCKER

```

### DOCKER DATEI INHALT

```

core@core-01 ~ $ cat nginx_servers/ng1/Dockerfile
FROM nginx
COPY static-html-directory /usr/share/nginx/html

```

## KONFIGURATION ANLAGE 10: CONTAINER 2 KONFIGURATIONEN

### NGINX ORDNER INHALT

```

core@core-01 ~ $ cat nginx_servers/ng1/index.html
nginx_2 DOCKER

```

### DOCKER DATEI INHALT

```

core@core-01 ~ $ cat nginx_servers/ng2/Dockerfile
FROM nginx
COPY static-html-directory /usr/share/nginx/html

```

## KONFIGURATION ANLAGE 11: CONTAINER 3 KONFIGURATIONEN

### NGINX ORDNER INHALT

```

core@core-01 ~ $ cat nginx_servers/ng3/index.html
nginx_3 DOCKER

```

### DOCKER DATEI INHALT

```

core@core-01 ~ $ cat nginx_servers/ng3/Dockerfile
FROM nginx

```

```
COPY static-html-directory /usr/share/nginx/html
```

## PROXY SYSTEM ANLAGEN

### PROXY ANLAGE 1: PROXY-SERVER DNS KONFIGURATIONEN

#### PROXY SERVER RNDC.KEY

---

##### GENERIERUNG VON RNDC.KEY

```
root@proxy:~# rndc-confgen
# Start of rndc.conf
key "rndc-key" {
    algorithm hmac-md5;
    secret "????????????????????";
};

options {
    default-key "rndc-key";
    default-server 127.0.0.1;
    default-port 953;
};
# End of rndc.conf

# Use with the following in named.conf, adjusting the allow list as needed:
# key "rndc-key" {
#     algorithm hmac-md5;
#     secret "????????????????????";
# };
#
# controls {
#     inet 127.0.0.1 port 953
#         allow { 127.0.0.1; } keys { "rndc-key"; };
# };
# End of named.conf
```

---

##### EINSETZUNG VON GENERIERTE RNDC.KEY SCHLÜSSEL

```
root@proxy:~# cat /etc/bind/rndc.key
key "rndc-key" {
    algorithm hmac-md5;
    secret "????????????????????";
};
```

---

##### TESTEN KONFIGURATIONEN VON RNDC.KEY

```
root@proxy:~# rndc status
version: 9.9.5-3ubuntu0.8-Ubuntu <id:f9b8a50e>
CPUs found: 1
worker threads: 1
UDP listeners per interface: 1
number of zones: 101
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
recursive clients: 0/0/1000
tcp clients: 0/100
server is up and running
```

---

#### PROXY SERVER DB.WEBPROXY.LOCAL.FORWARD DATEI

```
root@proxy:~# cat /etc/bind/db.webproxy.local.forward
;
; BIND data file for local loopback interface
;
$TTL 604800
@      IN      SOA  webproxy.local. root.webproxy.local. (
                        2          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
@      IN      NS   webproxy.local.
@      IN      A    172.16.2.2
@      IN      AAAA ::1
proxy  IN      A    172.16.2.2
webalizer IN    A    172.16.2.10
single      IN    A    172.16.2.11
round IN      A    172.16.2.12
weighted   IN    A    172.16.2.13
persistent IN    A    172.16.2.14
least              IN    A    172.16.2.15
;;
c-single      IN    A    172.16.2.21
c-round       IN    A    172.16.2.22
```

```
c-weighted    IN    A      172.16.2.23
c-persistent  IN    A      172.16.2.24
c-least       IN    A      172.16.2.25
```

---

#### PROXY SERVER DB.WEBPROXY.LOCAL.REV DATEI

```
root@proxy:~# cat /etc/bind/db.webproxy.local.rev

;
; BIND reverse data file for local loopback interface
;
$TTL 604800
;@      IN      SOA      localhost. root.localhost. (
@        IN      SOA      webproxy.local. root.webproxy.local. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache TTL
;
;@      IN      NS       localhost.
;1.0.0   IN      PTR     localhost.
@        IN      NS       webproxy.local
2        IN      PTR     webproxy.local
;; LOAD BALANCING
11       IN      PTR     single.webproxy.local
12       IN      PTR     round.webporxy.local
13       IN      PTR     weighted.webporxy.local
14       IN      PTR     persistent.webporxy.local
15       IN      PTR     least.webporxy.local
;; LOAD BALANCING FROM CONTAINERS
21       IN      PTR     c-single.webporxy.local
22       IN      PTR     c-round.webporxy.local
23       IN      PTR     c-weighted.webporxy.local
24       IN      PTR     c-persistent.webporxy.local
25       IN      PTR     c-least.webporxy.local
```

---

#### PROXY SERVER NAMED.CONF DATEI

```
root@proxy:~# cat /etc/bind/named.conf

// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
```



```
// this configuration file.
//
// If you are just adding zones, please do that in
/etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";

// Using rndc.key;
include "/etc/bind/rndc.key";
controls {
    inet 127.0.0.1 port 953 allow { localhost; } keys { "rndc-key"; };
};
```

---

#### PROXY SERVER NAMED.CONF.LOCAL DATEI

```
root@proxy:~# cat /etc/bind/named.conf.local
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

// FORWARD ZONE
zone "webproxy.local" {
    type master;
    file "/etc/bind/db.webproxy.local.forward";
};

//REVERSE ZONE
zone "2.16.173.in-addr.arpa" {
    type master;
    file "/etc/bind/db.webproxy.local.rev";
};
```

---

#### PROXY SERVER NAMED.CONF.OPTIONS DATEI

```
root@proxy:~# cat /etc/bind/named.conf.options
//MY ACLs
acl internals {
```

```

    localhost;
    localnets;
};

options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    // forwarders {
    //     0.0.0.0;
    // };

    // MY RESTRICTIONS
    allow-query {
        internals;
    };

    allow-recursion {
        internals;
    };

    recursion yes;

    allow-transfer {
        internals;
    };

    allow-update-forwarding {
        internals;
    };

    //=====

```

```
// If BIND logs error messages about the root key being expired,
// you will need to update your keys. See https://www.isc.org/bind-
keys
//=====
//dnssec-validation auto;
dnssec-enable yes;

auth-nxdomain no;    # conform to RFC1035
listen-on-v6 { any; };
listen-on port 53 {172.16.2.2;};
};
```

## PROXY ANLAGE 2: PROXY-SERVER BENUTZERRICHTLINIEN KONFIGURATIONEN

### EINSETZEN NEUE BENUTZER INS BENUTZERRICHTLINIEN

```
root@proxy:~# htpasswd /etc/nginx/htaccess/.htpasswd test_neue_benutzer
root@proxy:~# htpasswd /etc/nginx/htaccess/.htpasswd test_neue_benutzer
New password:
Re-type new password:
Adding password for user test_neue_benutzer
```

### INHALT HTACCES DATEI AUSDRÜCKT NACH NEUE BENUTZER EINSETZUNG

```
root@proxy:~# cat /etc/nginx/htaccess/.htpasswd
web:$apr1$I.hZ.f.P$BvecHqOYjDjydtxF4MXjU/
test_neue_benutzer:$apr1$H6EaVyeB$X4C8j5E85ZT3L0.JO6ZXD.
```

### BENUTZER LÖSCHUNG

```
root@proxy:~# htpasswd -D /etc/nginx/htaccess/.htpasswd test_neue_benutzer
Deleting password for user test_neue_benutzer
```

### INHALT HTACCES DATEI AUSDRÜCKT NACH BENUTZER LÖSCHUNG

```
root@proxy:~# cat /etc/nginx/htaccess/.htpasswd
web:$apr1$I.hZ.f.P$BvecHqOYjDjydtxF4MXjU/
```

## PROXY ANLAGE 3: WEBSERVER ALS PROXY BEISPIEL KONFIGURATION

```
root@proxy:~# cat /etc/nginx/sites-available/single
server {
    listen 172.16.2.11:80;
    server_name single.webproxy.local;

    location / {
        proxy_pass http://10.0.0.11:80;
        proxy_redirect default;
```

```

        proxy_read_timeout 900;
        proxy_send_timeout 900;

        ### HTACCESS: Enable User Login
        auth_basic "Restricted";
        auth_basic_user_file /etc/nginx/htaccess/.htpasswd;
    }
}

```

#### PROXY ANLAGE 4: ROUND-ROBIN KONFIGURATION

```

upstream web_round {
    server 10.0.0.11:80;
    server 10.0.0.22:80;
}

server {
    listen 172.16.2.12:80;
    server_name round.webproxy.local;
    location / {
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_pass http://web_round;
    }
}

```

#### PROXY ANLAGE 5: WEIGHTED KONFIGURATION.

```

upstream web_weighted {
    server 10.0.0.11:80 weight=3;
    server 10.0.0.22:80;
    server 10.0.0.33:80;
}

server {
    listen 172.16.2.13:80;
    server_name weighted.webproxy.local;
    location / {
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_pass http://web_weighted;
    }
}

```

#### PROXY ANLAGE 6: PERSISTENT KONFIGURATION

```

upstream web_persistent {
    ip_hash;
    server 10.0.0.11:80;
}

```

```

        server 10.0.0.22:80;
        server 10.0.0.33:80;
    }
    server {
        listen 172.16.2.14:80;
        server_name persistent.webproxy.local;
        location / {
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_pass http://web_persistent;
        }
    }
}

```

#### PROXY ANLAGE 7: LEAST-CONNECTED KONFIGURATION

```

upstream web_least {
    least_conn;
    server 10.0.0.11:80;
    server 10.0.0.22:80;
    server 10.0.0.33:80;
}

server {
    listen 172.16.2.15:80;
    server_name least.webproxy.local;
    location / {
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_pass http://web_least;
    }
}

```

#### PROXY ANLAGE 8: WEBALIZER WEBSEITE KONFIGURATION

```

root@proxy:~# cat /etc/nginx/sites-available/webalizer
server {
    listen 172.16.2.10:80;
    server_name webalizer.webproxy.local;

    location / {
        root /var/www/webalizer;
        index index.html;

        ### AUTH
        auth_basic 'Restricted';
        auth_basic_user_file /etc/nginx/htaccess/.webalizeraccess;
    }
}

```

```
}
```

## PROXY ANLAGE 9: FIREWALL

---

### DATEI

```
root@proxy:~# tree scripts/firewall/
scripts/firewall/
├── 10_containers_singles.sh
├── 11_containers_balancing.sh
├── 2_test_files.sh
├── 4_SPI.sh
├── 5_log.sh
├── 6_filters.sh
├── 7_servers_singles.sh
├── 8_servers_balancing.sh
├── 9_containers_ports.sh
├── containers
│   ├── containers_balancing
│   └── containers_singles
├── servers
│   ├── servers_balancing
│   └── servers_singles
└── variables
    ├── adapters
    ├── containers_ports
    ├── networks
    ├── servers
    └── used_ports
```

3 directories, 21 files

---

### FUNKTIONEN

Hier werden die Firewallfunktionen erklärt die im System benutzt werden.

---

#### FIREWALL-REGEL DEAKTIVIEREN

Alle existierenden Firewall Regeln werden gelöscht.

```
$fw -F
```

```
$fw -X
```

---

#### TEST EXISTING FILES

Die Prüfung ob alle Datei im System vorhanden sind.

```

files_exist=$(/bin/bash $DIRECTORY/2_test_files.sh $DIRECTORY)
if [ "$files_exist" ]
then
    echo ; echo -e "\e[1;31m File not found: $files_exist"; tput sgr0
    exit
else
    echo "All dependency files found"
fi

```

---

## OPEN ALL RULES

Hier werden alle Firewall Regeln eröffnet.

```

# Eigehend Offen
$fw -P INPUT DROP

# Ausgehend Offen
$fw -P OUTPUT DROP

# Routing erlauben
$fw -P FORWARD DROP

```

---

## STATE PACKET INSPEKTION

Hier werden die SPI eingesetzt.

```

#!/bin/bash

### Lokale Prozesse über localhost erlauben --> lo == localhost(127.0.0.1)
echo -e "\e[1;33m 4.2.1- SET SPI RULES: lo"; tput sgr0
$fw -A INPUT -i lo -j ACCEPT
$fw -A OUTPUT -o lo -j ACCEPT

### INPUT
echo -e "\e[1;33m 4.2.2- SET SPI RULES: INPUT"; tput sgr0
$fw -A INPUT -i $host_adapter -p tcp -m state --state ESTABLISHED,RELATED -j ACCEPT
$fw -A INPUT -i $to_server_adapter -p tcp -m state --state ESTABLISHED,RELATED -j ACCEPT
$fw -A INPUT -i $proxy_adapter -p tcp -m state --state ESTABLISHED,RELATED -j ACCEPT

### OUTPUT
echo -e "\e[1;33m 4.2.3- SET SPI RULES: OUTPUT"; tput sgr0
$fw -A OUTPUT -o $host_adapter -m state --state ESTABLISHED,RELATED -j ACCEPT
$fw -A OUTPUT -o $to_server_adapter -m state --state ESTABLISHED,RELATED -j ACCEPT

```

```
$fw -A OUTPUT -o $proxy_adapter -m state --state ESTABLISHED,RELATED -j  
ACCEPT
```

---

## LOG ACTIONS

Hier werden die Log funktionalitäten eingesetzt.

```
#!/bin/bash  
fw=/sbin/iptables  
### 5.2- CREATE LOG  
$fw -N DROPLOG  
### LOGGING ICMP:PING  
$fw -A DROPLOG -p icmp -m limit --limit 5/minute -j LOG --log-prefix "icmp-  
paket: "  
### LOGGING UDP  
$fw -A DROPLOG -p udp -m limit --limit 25/minute -j LOG --log-prefix "udp-  
paket: "  
### LOGGING TCP"  
$fw -A DROPLOG -p tcp -m limit --limit 50/minute -j LOG --log-prefix "tcp-  
paket: "  
### NOT LOGGING WITHOUT COMMENTS"  
$fw -A DROPLOG -j DROP  
### PORTSCANS BLOCK  
$fw -A INPUT -m limit --limit 5/minute -m state --state INVALID -j DROPLOG
```

---

## FILTERS

Hier werden alle Filter eingesetzt.

```
#!/bin/bash  
fw=/sbin/iptables  
DIRECTORY=$1  
source $DIRECTORY/variables/adapters  
source $DIRECTORY/variables/networks  
source $DIRECTORY/variables/used_ports  
  
### SET FILTER RULES  
  
### SSH  
$fw -A INPUT -p tcp --dport $port_ssh -i $to_server_adapter -s $lanB -d  
$to_server_ip -j ACCEPT  
# This is used to connect the host to this guest the Virtual-Machine  
$fw -A INPUT -p tcp --dport $port_ssh -i $host_adapter -s $host_ip -d  
$guest_ip -j ACCEPT  
  
### DNS
```



```

$fw -A INPUT -p udp --dport $port_dns -i $to_server_adapter -s $lanB -d
$to_server_ip -j ACCEPT

### PING

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanB -d
$to_server_ip -j ACCEPT

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanC -d
$to_server_ip -j ACCEPT

# This server can be pinged from lanA

$fw -A INPUT -p icmp --icmp-type 8 -i $proxy_adapter -s $lanA -d $proxy_ip
-j ACCEPT

### HTTP

$fw -A INPUT -p tcp --dport $port_http -i $proxy_adapter -s $lanA -d
$proxy_ip -j ACCEPT

$fw -A OUTPUT -p tcp --dport $port_http -o $proxy_adapter -s $proxy_ip -j
ACCEPT

```

---

## SERVER SINGLES

Hier werden alle Regeln eingesetzt ie nicht im Load-Balacing sind.

```

#!/bin/bash
fw=/sbin/iptables
DIRECTORY=$1
source $DIRECTORY/variables/adapters
source $DIRECTORY/variables/networks
source $DIRECTORY/variables/used_ports
source $DIRECTORY/servers/servers_singles

### PING

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanB -d
$single_ip -j ACCEPT

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanC -d
$single_ip -j ACCEPT

### HTTP

$fw -A INPUT -p tcp --dport $port_http -i $to_server_adapter -s $lanB -d
$single_ip -j ACCEPT

$fw -A INPUT -p tcp --dport $port_http -i $to_server_adapter -s $lanC -d
$single_ip -j ACCEPT

```

---

## SERVER LOAD-BALANCING

Hier werden alle Regeln eingesetzt die im Load-Balacing sind.

```

#!/bin/bash
### GET VARIABLES

```

```

fw=/sbin/iptables
DIRECTORY=$1
source $DIRECTORY/variables/adapters
source $DIRECTORY/variables/networks
source $DIRECTORY/variables/servers
source $DIRECTORY/variables/used_ports
source $DIRECTORY/servers/servers_balancing

### ROUND-ROBIN

# PING

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanB -d
$round_ip -j ACCEPT

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanC -d
$round_ip -j ACCEPT

# HTTP

$fw -A INPUT -p tcp --dport $port_http -i $to_server_adapter -s $lanB -d
$round_ip -j ACCEPT

$fw -A INPUT -p tcp --dport $port_http -i $to_server_adapter -s $lanC -d
$round_ip -j ACCEPT


### WEIGHTED

# PING

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanB -d
$weighted_ip -j ACCEPT

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanC -d
$weighted_ip -j ACCEPT

# HTTP

$fw -A INPUT -p tcp --dport $port_http -i $to_server_adapter -s $lanB -d
$weighted_ip -j ACCEPT

$fw -A INPUT -p tcp --dport $port_http -i $to_server_adapter -s $lanC -d
$weighted_ip -j ACCEPT


### PERSISTENT

# PING

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanB -d
$persistent_ip -j ACCEPT

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanC -d
$persistent_ip -j ACCEPT

# HTTP

$fw -A INPUT -p tcp --dport $port_http -i $to_server_adapter -s $lanB -d
$persistent_ip -j ACCEPT

$fw -A INPUT -p tcp --dport $port_http -i $to_server_adapter -s $lanC -d
$persistent_ip -j ACCEPT


### LEAST-CONNECTED

```

```
# PING

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanB -d
$least_conn_ip -j ACCEPT

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanC -d
$least_conn_ip -j ACCEPT

# HTTP

$fw -A INPUT -p tcp --dport $port_http -i $to_server_adapter -s $lanB -d
$least_conn_ip -j ACCEPT

$fw -A INPUT -p tcp --dport $port_http -i $to_server_adapter -s $lanC -d
$least_conn_ip -j ACCEPT
```

---

## CONTAINERS PORTS

Hier sind die Regeln für die Container die in der Testumgebung vorhanden sind.

```
#!/bin/bash

### GET VARIABLES

fw=/sbin/iptables

DIRECTORY=$1

source $DIRECTORY/variables/adapters
source $DIRECTORY/variables/networks
source $DIRECTORY/variables/containers_ports

### CONTAINER 1

$fw -A INPUT -p tcp --dport $container1_port -i $proxy_adapter -s $lanA -d
$proxy_ip -j ACCEPT

$fw -A OUTPUT -p tcp --dport $container1_port -o $proxy_adapter -s
$proxy_ip -j ACCEPT

### CONTAINER 2

$fw -A INPUT -p tcp --dport $container2_port -i $proxy_adapter -s $lanA -d
$proxy_ip -j ACCEPT

$fw -A OUTPUT -p tcp --dport $container2_port -o $proxy_adapter -s
$proxy_ip -j ACCEPT

### CONTAINER 3

$fw -A INPUT -p tcp --dport $container3_port -i $proxy_adapter -s $lanA -d
$proxy_ip -j ACCEPT

$fw -A OUTPUT -p tcp --dport $container3_port -o $proxy_adapter -s
$proxy_ip -j ACCEPT
```

---

## CONTAINERS SINGLES

Hier sind die Regeln für die Container die in nicht im Load-Balancing sind.

```
#!/bin/bash

###GET VARIABLES"; tput sgr0

fw=/sbin/iptables

DIRECTORY=$1

source $DIRECTORY/variables/adapters
```

```

source $DIRECTORY/variables/networks
source $DIRECTORY/variables/containers_ports
source $DIRECTORY/containers/containers_singles

### D-SINGLE

# PING

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanB -d
$d_single_ip -j ACCEPT

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanC -d
$d_single_ip -j ACCEPT

# HTTP

$fw -A INPUT -p tcp --dport 80 -i $to_server_adapter -s $lanB -d
$d_single_ip -j ACCEPT

$fw -A INPUT -p tcp --dport 80 -i $to_server_adapter -s $lanC -d
$d_single_ip -j ACCEPT

```

---

## CONTAINERS LOAD-BALANCING

Hier sind die Regeln für die Container die im Load-Balancing sind.

```

/bin/bash $DIRECTORY/11_containers_balancing.sh $DIRECTORY
AAA
#!/bin/bash
### GET VARIABLES"; tput sgr0
fw=/sbin/iptables
DIRECTORY=$1

source $DIRECTORY/variables/adapters
source $DIRECTORY/variables/networks
source $DIRECTORY/variables/containers_ports
source $DIRECTORY/containers/containers_balancing

### D-ROUND

# PING

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanB -d
$d_round_ip -j ACCEPT

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanC -d
$d_round_ip -j ACCEPT

# HTTP

$fw -A INPUT -p tcp --dport 80 -i $to_server_adapter -s $lanB -d
$d_round_ip -j ACCEPT

$fw -A INPUT -p tcp --dport 80 -i $to_server_adapter -s $lanC -d
$d_round_ip -j ACCEPT

### D-WEIGHTED

# PING

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanB -d
$d_weighted_ip -j ACCEPT

```

```

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanC -d
$d_weighted_ip -j ACCEPT

# HTTP

$fw -A INPUT -p tcp --dport 80 -i $to_server_adapter -s $lanB -d
$d_weighted_ip -j ACCEPT

$fw -A INPUT -p tcp --dport 80 -i $to_server_adapter -s $lanC -d
$d_weighted_ip -j ACCEPT

### D-PERSISTENT

# PING

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanB -d
$d_persistent_ip -j ACCEPT

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanC -d
$d_persistent_ip -j ACCEPT

# HTTP

$fw -A INPUT -p tcp --dport 80 -i $to_server_adapter -s $lanB -d
$d_persistent_ip -j ACCEPT

$fw -A INPUT -p tcp --dport 80 -i $to_server_adapter -s $lanC -d
$d_persistent_ip -j ACCEPT

### D-LEAST

# PING

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanB -d
$d_least_conn_ip -j ACCEPT

$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanC -d
$d_least_conn_ip -j ACCEPT

# HTTP

$fw -A INPUT -p tcp --dport 80 -i $to_server_adapter -s $lanB -d
$d_least_conn_ip -j ACCEPT

$fw -A INPUT -p tcp --dport 80 -i $to_server_adapter -s $lanC -d
$d_least_conn_ip -j ACCEPT

```

---

## IP TABLES OUTPUT

Alle Firewall Regeln und Ketten nach dem die Konfiguration abgeschlossen wurde.

```

root@proxy:~# iptables -L

Chain INPUT (policy DROP)

target      prot opt source                destination
ACCEPT      all  --  anywhere              anywhere
ACCEPT      tcp  --  anywhere              anywhere             state RELATED,ESTABLISHED
ACCEPT      tcp  --  anywhere              anywhere             state RELATED,ESTABLISHED
ACCEPT      tcp  --  anywhere              anywhere             state RELATED,ESTABLISHED
ACCEPT      tcp  --  172.16.2.0/24         172.16.2.2          tcp dpt:ssh
ACCEPT      tcp  --  192.168.1.104         192.168.1.107       tcp dpt:ssh
ACCEPT      udp  --  172.16.2.0/24         172.16.2.2          udp dpt:domain
ACCEPT      icmp --  172.16.2.0/24         172.16.2.2          icmp echo-request
ACCEPT      icmp --  192.168.3.0/24        172.16.2.2          icmp echo-request
ACCEPT      icmp --  10.0.0.0/24           proxy.webproxy.local icmp echo-request

```

ACCEPT	tcp	--	10.0.0.0/24	proxy.webproxy.local	tcp dpt:http
ACCEPT	icmp	--	172.16.2.0/24	172.16.2.11	icmp echo-request
ACCEPT	icmp	--	192.168.3.0/24	172.16.2.11	icmp echo-request
ACCEPT	tcp	--	172.16.2.0/24	172.16.2.11	tcp dpt:http
ACCEPT	tcp	--	192.168.3.0/24	172.16.2.11	tcp dpt:http
ACCEPT	icmp	--	172.16.2.0/24	172.16.2.12	icmp echo-request
ACCEPT	icmp	--	192.168.3.0/24	172.16.2.12	icmp echo-request
ACCEPT	tcp	--	172.16.2.0/24	172.16.2.12	tcp dpt:http
ACCEPT	tcp	--	192.168.3.0/24	172.16.2.12	tcp dpt:http
ACCEPT	icmp	--	172.16.2.0/24	172.16.2.13	icmp echo-request
ACCEPT	icmp	--	192.168.3.0/24	172.16.2.13	icmp echo-request
ACCEPT	tcp	--	172.16.2.0/24	172.16.2.13	tcp dpt:http
ACCEPT	tcp	--	192.168.3.0/24	172.16.2.13	tcp dpt:http
ACCEPT	icmp	--	172.16.2.0/24	172.16.2.14	icmp echo-request
ACCEPT	icmp	--	192.168.3.0/24	172.16.2.14	icmp echo-request
ACCEPT	tcp	--	172.16.2.0/24	172.16.2.14	tcp dpt:http
ACCEPT	tcp	--	192.168.3.0/24	172.16.2.14	tcp dpt:http
ACCEPT	icmp	--	172.16.2.0/24	172.16.2.15	icmp echo-request
ACCEPT	icmp	--	192.168.3.0/24	172.16.2.15	icmp echo-request
ACCEPT	tcp	--	172.16.2.0/24	172.16.2.15	tcp dpt:http
ACCEPT	tcp	--	192.168.3.0/24	172.16.2.15	tcp dpt:http
ACCEPT	tcp	--	10.0.0.0/24	proxy.webproxy.local	tcp dpt:4001
ACCEPT	tcp	--	10.0.0.0/24	proxy.webproxy.local	tcp dpt:4002
ACCEPT	tcp	--	10.0.0.0/24	proxy.webproxy.local	tcp dpt:4003
ACCEPT	icmp	--	172.16.2.0/24	172.16.2.21	icmp echo-request
ACCEPT	icmp	--	192.168.3.0/24	172.16.2.21	icmp echo-request
ACCEPT	tcp	--	172.16.2.0/24	172.16.2.21	tcp dpt:http
ACCEPT	tcp	--	192.168.3.0/24	172.16.2.21	tcp dpt:http
ACCEPT	icmp	--	172.16.2.0/24	172.16.2.22	icmp echo-request
ACCEPT	icmp	--	192.168.3.0/24	172.16.2.22	icmp echo-request
ACCEPT	tcp	--	172.16.2.0/24	172.16.2.22	tcp dpt:http
ACCEPT	tcp	--	192.168.3.0/24	172.16.2.22	tcp dpt:http
ACCEPT	icmp	--	172.16.2.0/24	172.16.2.23	icmp echo-request
ACCEPT	icmp	--	192.168.3.0/24	172.16.2.23	icmp echo-request
ACCEPT	tcp	--	172.16.2.0/24	172.16.2.23	tcp dpt:http
ACCEPT	tcp	--	192.168.3.0/24	172.16.2.23	tcp dpt:http
ACCEPT	icmp	--	172.16.2.0/24	172.16.2.24	icmp echo-request
ACCEPT	icmp	--	192.168.3.0/24	172.16.2.24	icmp echo-request
ACCEPT	tcp	--	172.16.2.0/24	172.16.2.24	tcp dpt:http
ACCEPT	tcp	--	192.168.3.0/24	172.16.2.24	tcp dpt:http
ACCEPT	icmp	--	172.16.2.0/24	172.16.2.25	icmp echo-request
ACCEPT	icmp	--	192.168.3.0/24	172.16.2.25	icmp echo-request
ACCEPT	tcp	--	172.16.2.0/24	172.16.2.25	tcp dpt:http
ACCEPT	tcp	--	192.168.3.0/24	172.16.2.25	tcp dpt:http

Chain FORWARD (policy DROP)

target	prot	opt	source	destination
--------	------	-----	--------	-------------

Chain OUTPUT (policy DROP)

target	prot	opt	source	destination	
ACCEPT	all	--	anywhere	anywhere	
ACCEPT	all	--	anywhere	anywhere	state RELATED,ESTABLISHED
ACCEPT	all	--	anywhere	anywhere	state RELATED,ESTABLISHED
ACCEPT	all	--	anywhere	anywhere	state RELATED,ESTABLISHED
ACCEPT	tcp	--	proxy.webproxy.local	anywhere	tcp dpt:http
ACCEPT	tcp	--	proxy.webproxy.local	anywhere	tcp dpt:4001
ACCEPT	tcp	--	proxy.webproxy.local	anywhere	tcp dpt:4002
ACCEPT	tcp	--	proxy.webproxy.local	anywhere	tcp dpt:4003

Chain DROPLOG (0 references)

target	prot	opt	source	destination	
LOG	icmp	--	anywhere	anywhere	limit: avg 5/min burst 5 LOG
level warning prefix "icmp-paket: "					
LOG	udp	--	anywhere	anywhere	limit: avg 25/min burst 5 LOG
level warning prefix "udp-paket: "					
LOG	tcp	--	anywhere	anywhere	limit: avg 50/min burst 5 LOG
level warning prefix "tcp-paket: "					
DROP	all	--	anywhere	anywhere	

## PROXY ANLAGE 10: PROXY SYSTEM

DATEI

```
root@proxy:~# tree ~/scripts/server_proxy/
/root/scripts/server_proxy/
├─ added_containers
├─ added_servers
├─ containers
│   ├─ container_balancing_del.sh
│   ├─ container_balancing_menu.sh
│   ├─ container_balancing_new.sh
│   ├─ container_balancing_set.sh
│   ├─ del_containers.sh
│   ├─ menu_containers.sh
│   ├─ new_containers.sh
│   ├─ show_containers.sh
│   ├─ single_container_del.sh
│   ├─ single_container_new.sh
│   └─ single_container_set.sh
├─ servers
│   ├─ del_servers.sh
│   └─ load_balancing_del.sh
```

- |   └─ load\_balancing\_menu.sh
- |   └─ load\_balancing\_new.sh
- |   └─ load\_balancing\_set.sh
- |   └─ menu\_servers.sh
- |   └─ new\_servers.sh
- |   └─ show\_servers.sh
- |   └─ single\_server\_del.sh
- |   └─ single\_server\_new.sh
- |   └─ single\_server\_set.sh
- └─ shared
  - |   └─ dns\_entry\_del.sh
  - |   └─ dns\_entry\_new.sh
  - |   └─ fw\_single\_del.sh
  - |   └─ fw\_single\_new.sh
  - |   └─ interface\_del.sh
  - |   └─ interface\_new.sh
  - |   └─ proxy\_balancing\_new.sh
  - |   └─ proxy\_single\_del.sh
  - |   └─ proxy\_single\_new.sh
- └─ users
  - |   └─ del\_users.sh
  - |   └─ menu\_users.sh
  - |   └─ new\_users.sh
  - |   └─ show\_users.sh
  - |   └─ update\_users.sh
- └─ variables
  - |   └─ adapter\_number
  - |   └─ balancing\_addresses
  - |   └─ balancing\_counters
  - |   └─ domain\_name
  - |   └─ new\_ip

7 directories, 41 files

---

## FLUSSDIAGRAM

Siehe Anlage Ende Dokument

---

## FUNKTIONEN

Hier werden die Funktionen beschrieben, die im System vorhanden sind.



---

## ADDED\_CONTAINERS ORDNER

Hier werden die Konfigurationdatei von neue Container eingesetzt.

---

## ADDED\_SERVERS ORDNER

Hier werden die Konfigurationdatei von neue Server eingesetzt.

---

## CONTAINERS ORDNER

---

### CONTAINER\_BALANCING\_DEL.SH DATEI

---

```
#!/bin/bash

DIRECTORY=/root/scripts/server_proxy

### LOAD FROM VARIABLES
source $DIRECTORY/variables/domain_name
source $DIRECTORY/variables/balancing_counters

### READ SERVER NAME
read -p "Load-Balancing Container name to be deleted: " server_name
echo -e "Container name: \e[1;31m$server_name"; tput sgr0
read -p "Is that correct(Y/N): " option_server_name

if [ $option_server_name == "Y" ] || [ $option_server_name == "y" ]
then

    entry=$DIRECTORY/added_containers/$server_name

    if [ -f $entry ]
    then
        source $entry

        if [ $type == load-balancing ]
        then
            echo ""; echo -e "\e[1;31m REMOVE CONTAINER FROM BALANCING"; tput
sgr0

            ### STRIP
            proxy_file="${proxy_address%}.${domain_name}"

            ### UPDATE
            sed -i "/server $new_server_ip:$new_server_port/d" /etc/nginx/sites-
available/$proxy_file

            echo ""; echo -e "\e[1;31m UPDATE BALANCING VARIABLE FILE"; tput sgr0
```

```

### GET VALUES
counter_name=${proxy_file}"_count"
new_counter=$((counter_name-1))
### UPDATE
sed -i "/$counter_name/ c $counter_name=$new_counter"
$DIRECTORY/variables/balancing_counters

echo "" ; echo -e "\e[1;31m DELETE SERVER CONFIG FILE"; tput sgr0
rm -rfv $DIRECTORY/added_containers/$server_name

echo "" ; echo -e "\e[1;33m RESTART PROXY"; tput sgr0
service nginx restart
else
    echo "Server chosen is not Load-Balancing..."
fi
else
    echo "server_name not found :("
fi
else
    echo "Aborted by choosing single server name..."
fi

```

#### CONTAINER\_BALANCING\_MENU.SH DATEI

---

```

#!/bin/bash
DIRECTORY=/root/scripts/server_proxy
while [ option != "0" ]
do
    clear;echo
    echo -e "\e[31mADD CONTAINER TO LOAD-BALANCING MENU"; tput sgr0
    echo ""
    echo "1          Add to Round-Robin"
    echo "2          Add to Weighted"
    echo "3          Add to Persistent"
    echo "4          Add to Least Connected"
    echo ""
    read -p "Option: " option
    case $option in
    1)
        echo "Adding to Round-Robin..."
        /bin/bash $DIRECTORY/containers/container_balancing_new.sh "c-round"
        break;;

```

```

2)
    echo "Adding to Weighted..."
    /bin/bash $DIRECTORY/containers/container_balancing_new.sh "c-
weighted"
    break;;
3)
    echo "Adding to Persistent..."
    /bin/bash $DIRECTORY/containers/container_balancing_new.sh "c-
persistent"
    break;;
4)
    echo "Adding to Least Connected..."
    /bin/bash $DIRECTORY/containers/container_balancing_new.sh "c-least"
    break;;
*)
    echo "Not a valid selection in CONTAINER LOAD-BALANCING MENU"
    break;;
esac
echo "Press ENTER to go back to MENU";
done

```

#### CONTAINER\_BALANCING\_NEW.SH DATEI

---

```

#!/bin/bash
DIRECTORY=/root/scripts/server_proxy
### GET SENT VARIABLES
balancing_type=$1

### LOAD FROM VARIABLES
source $DIRECTORY/variables/domain_name

### READ BALANCING NAME
proxy_address=${balancing_type}".$domain_name"
echo "" ; echo -e "Server to be added to: \e[1;31m$proxy_address"; tput
sgr0
read -p "Is that correct(Y/N): " option_adding

if [ $option_adding == "Y" ] || [ $option_adding == "y" ]
then
    read -p "Please enter the Container-Server IP adress: "
balancing_server_ip
    echo -e "Container-Server server IP: \e[1;31m$balancing_server_ip"; tput
sgr0

```

```

read -p "Is that correct(Y/N): " option_balancing_ip

if [ $option_balancing_ip == "Y" ] || [ $option_balancing_ip == "y" ]
then
    read -p "Please enter the Container port number: " container_port
    echo -e "Container port: \e[1;31m$container_port"; tput sgr0
    read -p "Is that correct(Y/N): " option_container_port

    if [ $option_container_port == "Y" ] || [ $option_container_port == "y"
]
    then
        /bin/bash $DIRECTORY/containers/container_balancing_set.sh
$balancing_type $balancing_server_ip $proxy_address $container_port
    else
        echo "Aborted by choosing Container port..."
    fi
else
    echo "Aborted by choosing load-balancing ip..."
fi
else
    echo "Aborted by choosing load-balancing server..."
fi

```

#### DONTAINER\_BALANCING\_SET.SH DATEI

```

#!/bin/bash

DIRECTORY=/root/scripts/server_proxy

### GET SENT VARIABLES

balancing_type=$1
balancing_server_ip=$2
proxy_address=$3
container_port=$4

### LOAD FROM VARIABLES

source $DIRECTORY/variables/balancing_counters
source $DIRECTORY/variables/balancing_addresses

### SET VARIABLES

case $balancing_type in
    "c-round")
        config_file_name=${balancing_type}"_$_c_round_count"
        proxy_ip=$_c_round_address

```

```

        new_counter=$((c_round_count+1))
        counter_name=c_round_count
    ;;
"c-weighted")
    config_file_name=${balancing_type}"_$_c_weighted_count"
    proxy_ip=$_c_weighted_address
    new_counter=$((c_weighted_count+1))
    counter_name=c_weighted_count
    ;;
"c-persistent")
    config_file_name=${balancing_type}"_$_c_persistent_count"
    proxy_ip=$_c_persistent_address
    new_counter=$((c_persistent_count+1))
    counter_name=c_persistent_count
    ;;
"c-least")
    config_file_name=${balancing_type}"_$_c_least_count"
    proxy_ip=$_c_least_address
    new_counter=$((c_least_count+1))
    counter_name=c_least_count
    ;;
esac

### TEST
echo "
c_round_count: $_c_round_count
config_file_name: $config_file_name
proxy_ip: $_proxy_ip
new_counter: $_new_counter
counter_name: $_counter_name
"

echo ""; echo -e "\e[1;31m ADD SERVER TO BALANCING: $_balancing_type"; tput
sgr0

/bin/bash $DIRECTORY/shared/proxy_balancing_new.sh $_balancing_type
$_balancing_server_ip $_container_port

echo ""; echo -e "\e[1;31m UPDATE BALANCING VARIABLE FILE"; tput sgr0
sed -i "$counter_name/ c $counter_name=$_new_counter"
$DIRECTORY/variables/balancing_counters

```

```

echo ""; echo -e "\e[1;31m CREATE SERVER CONFIG FILE"; tput sgr0
echo "server_name=$config_file_name
type=load-balancing
new_server_ip=$balancing_server_ip
new_server_port=$container_port
proxy_server_ip=$proxy_ip
proxy_address=$proxy_address" >
$DIRECTORY/added_containers/$config_file_name

```

```

echo " " ; echo -e "\e[1;33m RESTART PROXY"; tput sgr0
service nginx restart

```

---

#### DEL\_CONTAINER.SH DATEI

```

#!/bin/bash
DIRECTORY=/root/scripts/server_proxy/containers
while [ option != "0" ]
do
    clear;echo
    echo -e "\e[31mSELECT CONTAINER TYPE TO DELETE MENU"; tput sgr0
    echo ""
    echo "1          Delete Single Server Container"
    echo "2          Delete Container from Load-Balancing"
    echo ""
    read -p "Option: " option
    case $option in
        1)
            echo "Delete Single Container..."
            /bin/bash $DIRECTORY/single_container_del.sh
            break;;
        2)
            echo "Delete Container from Load-Balancing..."
            /bin/bash $DIRECTORY/container_balancing_del.sh
            break;;
        *)
            echo "Not a valid selection in CONTAINER TYPE TO DELETE MENU"
            break;;
    esac
    echo "Press ENTER to go back to MENU";
done

```

---

#### MENU\_CONTAINERS.SH DATEI

```

#!/bin/bash

```

```

DIRECTORY=/root/scripts/server_proxy/containers
while [ option != "0" ]
do
    clear;echo
    echo -e "\e[31mCONTAINERS MENU"; tput sgr0
    echo ""
    echo "1          Show Containers"
    echo "2          Add Container"
    echo "3          Delete Container"
    echo ""
    read -p "Option: " option

    case $option in
    1)
        echo "Show Containers..."
        /bin/bash $DIRECTORY/show_containers.sh
        break;;
    2)
        echo "Adding Container..."
        /bin/bash $DIRECTORY/new_containers.sh
        break;;
    3)
        echo "Deleting Container..."
        /bin/bash $DIRECTORY/del_containers.sh
        break;;
    *)
        echo "Not a valid selection in CONTAINERS MENU"
        break;;
    esac

    echo "Press ENTER to go back to MENU";
done

```

#### NEW\_CONTAINERS.SH DATEI

---

```

#!/bin/bash
DIRECTORY=/root/scripts/server_proxy/containers
while [ option != "0" ]
do
    clear;echo
    echo -e "\e[31mSELECT CONTAINER TYPE MENU"; tput sgr0
    echo ""

```

```

echo "1          Add as Single Server"
echo "2          Add Container to Load-Balancing"
echo ""
read -p "Option: " option
    case $option in
    1)
        echo "Adding as Single Server..."
        /bin/bash $DIRECTORY/single_container_new.sh
        break;;
    2)
        echo "Adding Container to Load-Balancing..."
        /bin/bash $DIRECTORY/container_balancing_menu.sh
        break;;
    *)
        echo "Not a valid selection in CONTAINERS TYPE MENU"
        break;;
    esac
echo "Press ENTER to go back to MENU";
done

```

#### SHOW\_CONTAINER.SH DATEI

---

```

#!/bin/bash
DIRECTORY=/root/scripts/server_proxy/added_containers
echo ""
if [ "$(ls -A $DIRECTORY)" ]
then
    for entry in "$DIRECTORY"/*
    do
        source $entry
        echo -e "CONTAINER_NAME: \e[1;33m$server_name"; tput sgr0
        echo -e "          TYPE: \e[1;32m$type"; tput sgr0
        echo "          IP: $new_server_ip"
        echo "          PORT: $new_server_port"
        echo "          PROXY: $proxy_server_ip"
        if [ $type == single_container ]
        then
            echo "          ADAPTER: $adapter_name"
        else
            echo "          ADDRESS: $proxy_address"
        fi
    done
done

```



```

else
    echo "No added servers found..."
fi

```

#### SINGLE\_CONTAINER\_DEL.SH DATEI

---

```

#!/bin/bash
DIRECTORY=/root/scripts/server_proxy
### LOAD FROM FILES
source $DIRECTORY/variables/domain_name
### READ SERVER NAME
read -p "Single Server container name to be deleted: " container_name
full_name=${container_name}".$domain_name"
echo "" ; echo -e "Container Server address: \e[1;31m$full_name"; tput sgr0
read -p "Is that correct(Y/N): " option_containter_name

if [ $option_containter_name == "Y" ] || [ $option_containter_name == "y" ]
then
    entry=$DIRECTORY/added_containers/$container_name

    if [ -f $entry ]
    then
        source $entry

        if [ $type == single_container ]
        then
            echo ""; echo -e "\e[1;31m DELETE NETWORK INTERFACE"; tput sgr0
            /bin/bash $DIRECTORY/shared/interface_del.sh $adapter_name

            echo ""; echo -e "\e[1;31m UPDATE DNS"; tput sgr0
            /bin/bash $DIRECTORY/shared/dns_entry_del.sh $server_name
$new_server_ip $domain_name

            echo "" ; echo -e "\e[1;31m UPDATE PROXY"; tput sgr0
            /bin/bash $DIRECTORY/shared/proxy_single_del.sh $server_name
$full_name

            echo "" ; echo -e "\e[1;31m UPDATE FIREWALL"; tput sgr0
            /bin/bash $DIRECTORY/shared/fw_single_del.sh $server_name
$new_server_ip

            echo "" ; echo -e "\e[1;31m DELETE SERVER CONFIG FILE"; tput sgr0
            rm -rfv $DIRECTORY/added_containers/$server_name

```

```

        else
            echo "Container chosen is not a Single Server..."
        fi
    else
        echo "Container name not found :("
    fi
else
    echo "Aborted by choosing single server container name..."
fi

```

#### SINGLE\_CONTAINER\_NEW.SH DATEI

---

```

#!/bin/bash

DIRECTORY=/root/scripts/server_proxy

### LOAD FROM FILES
source $DIRECTORY/variables/domain_name

### READ SERVER NAME
read -p "Single Container name to be added: " container_name
full_name=${container_name}".$domain_name"
echo -e "New server name: \e[1;31m$full_name"; tput sgr0
read -p "Is that correct(Y/N): " option_container_name

if [ $option_container_name == "Y" ] || [ $option_container_name == "y" ]
then
    read -p "Please enter the Container-Server IP adress: "
    container_server_ip

    echo -e "Container-Server IP: \e[1;31m$container_server_ip"; tput sgr0
    read -p "Is that correct(Y/N): " option_container_server_ip

    if [ $option_container_server_ip == "Y" ] || [
$option_container_server_ip == "y" ]
    then
        read -p "Please enter the Container port number: " container_port
        echo -e "Container port: \e[1;31m$container_port"; tput sgr0
        read -p "Is that correct(Y/N): " option_container_port

        if [ $option_container_port == "Y" ] || [ $option_container_port == "y"
]
        then
            /bin/bash $DIRECTORY/containers/single_container_set.sh
            $container_name $container_server_ip $container_port $domain_name
            $full_name
        else

```

```

        echo "Aborted by choosing Container port..."
    fi
else
    echo "Aborted by choosing Container-Server ip..."
fi
else
    echo "Aborted by choosing container server name..."
fi

```

#### SINGLE\_CONTAINER\_SET.SH DATEI

---

```

#!/bin/bash
DIRECTORY=/root/scripts/server_proxy
### GET SENT VARIABLES
$container_server_ip $container_port $domain_name $full_name
container_name=$1
container_server_ip=$2
container_port=$3
domain_name=$4
full_name=$5
### LOAD FROM FILES
source $DIRECTORY/variables/new_ip
source $DIRECTORY/variables/adapter_number
### SET VARIABLES
new_server_ip=172.16.2.$server_ip
adapter_name=eth1:added$adapter_number

echo "" ; echo -e "\e[1;31m NEW NETWORK INTERFACE"; tput sgr0
/bin/bash $DIRECTORY/shared/interface_new.sh $new_server_ip $adapter_name

echo "" ; echo -e "\e[1;31m UPDATE DNS"; tput sgr0
/bin/bash $DIRECTORY/shared/dns_entry_new.sh $container_name $new_server_ip
$server_ip $domain_name $full_name

echo "" ; echo -e "\e[1;31m UPDATE PROXY"; tput sgr0
/bin/bash $DIRECTORY/shared/proxy_single_new.sh $container_name
$new_server_ip $container_server_ip $full_name $container_port

echo "" ; echo -e "\e[1;31m UPDATE FIREWALL"; tput sgr0
/bin/bash $DIRECTORY/shared/fw_single_new.sh $container_name $new_server_ip

echo ""; echo -e "\e[1;31m CREATE CONTAINER CONFIG FILE"; tput sgr0

```

```

echo "server_name=$container_name
type=single_container
new_server_ip=$new_server_ip
proxy_server_ip=$container_server_ip:$container_port
adapter_name=$adapter_name" > $DIRECTORY/added_containers/$container_name

```

---

## SERVICES ORDER

### DEL\_SERVER.SH DATEI

---

```

#!/bin/bash
DIRECTORY=/root/scripts/server_proxy/servers
while [ option != "0" ]
do
    clear;echo
    echo -e "\e[31mSELECT SERVER TYPE MENU TO DELETE"; tput sgr0
    echo ""
    echo "1          Delete Single Server"
    echo "2          Delete Server from Load-Balancing"
    echo ""
    read -p "Option: " option
    case $option in
        1)
            echo "Delete Single Server..."
            /bin/bash $DIRECTORY/single_server_del.sh
            break;;
        2)
            echo "Delete Server from Load-Balancing..."
            /bin/bash $DIRECTORY/load_balancing_del.sh
            break;;
        *)
            echo "Not a Valid Selection at SERVER TYPE MENU TO DELETE"
            break;;
    esac
    echo "Press ENTER to go back to MENU";
done

```

### LOAD\_BALANCING\_DEL.SH DATEI

---

```

#!/bin/bash
DIRECTORY=/root/scripts/server_proxy
### LOAD FROM VARIABLES
source $DIRECTORY/variables/domain_name
source $DIRECTORY/variables/balancing_counters

```

```

### READ SERVER NAME
read -p "Load-Balancing Server name to be deleted: " server_name
echo -e "Server name: \e[1;31m$server_name"; tput sgr0
read -p "Is that correct(Y/N): " option_server_name

if [ $option_server_name == "Y" ] || [ $option_server_name == "y" ]
then

    entry=$DIRECTORY/added_servers/$server_name

    if [ -f $entry ]
    then
        source $entry

        if [ $type == load-balancing ]
        then
            echo ""; echo -e "\e[1;31m REMOVE SERVER FROM BALANCING"; tput sgr0
            ### STRIP
            proxy_file="{proxy_address%.$domain_name}"
            ### UPDATE
            sed -i "/server $new_server_ip:80/d" /etc/nginx/sites-
available/$proxy_file

            echo ""; echo -e "\e[1;31m UPDATE BALANCING VARIABLE FILE"; tput sgr0
            ### GET VALUES
            counter_name="{proxy_file}"_count"
            new_counter=$((counter_name-1))
            ### UPDATE
            sed -i "/$counter_name/ c $counter_name=$new_counter"
$DIRECTORY/variables/balancing_counters

            echo "" ; echo -e "\e[1;31m DELETE SERVER CONFIG FILE"; tput sgr0
            rm -rfv $DIRECTORY/added_servers/$server_name

            echo "" ; echo -e "\e[1;33m RESTART PROXY"; tput sgr0
            service nginx restart
        else
            echo "Server chosen is not Load-Balancing..."
        fi
    else

```

```

        echo "server_name not found :("
    fi
else
    echo "Aborted by choosing single server name..."
fi

```

#### LOAD\_BALANCING\_MENU.SH DATEI

---

```

#!/bin/bash
DIRECTORY=/root/scripts/server_proxy
while [ option != "0" ]
do
    clear;echo
    echo -e "\e[31mADD TO LOAD-BALANCING MENU"; tput sgr0
    echo ""
    echo "1          Add to Round-Robin"
    echo "2          Add to Weighted"
    echo "3          Add to Persistent"
    echo "4          Add to Least Connected"
    echo ""
    read -p "Option: " option
    case $option in
        1)
            echo "Adding to Round-Robin..."
            /bin/bash $DIRECTORY/servers/load_balancing_new.sh "round"
            break;;
        2)
            echo "Adding to Weighted..."
            /bin/bash $DIRECTORY/servers/load_balancing_new.sh "weighted"
            break;;
        3)
            echo "Adding to Persistent..."
            /bin/bash $DIRECTORY/servers/load_balancing_new.sh "persistent"
            break;;
        4)
            echo "Adding to Least Connected..."
            /bin/bash $DIRECTORY/servers/load_balancing_new.sh "least"
            break;;
        *)
            echo "Not a Valid Selection at LOAD-BALANCING MENU"
            break;;
    esac
done

```

```
    echo "Press ENTER to go back to MENU";
done
```

#### LOAD\_BALANCING\_NEW.SH DATEI

---

```
#!/bin/bash
DIRECTORY=/root/scripts/server_proxy
### GET SENT VARIABLES
balancing_type=$1
### LOAD FROM VARIABLES
source $DIRECTORY/variables/domain_name
### READ BALANCING NAME
proxy_address=${balancing_type}".$domain_name"
echo "" ; echo -e "Server to be added to: \e[1;31m$proxy_address"; tput sgr0
read -p "Is that correct(Y/N): " option_adding
if [ $option_adding == "Y" ] || [ $option_adding == "y" ]
then
    read -p "Please enter the IP adress: " balancing_server_ip
    echo -e "New server IP: \e[1;31m$balancing_server_ip"; tput sgr0
    read -p "Is that correct(Y/N): " option_balancing_ip

    if [ $option_balancing_ip == "Y" ] || [ $option_balancing_ip == "y" ]
    then
        /bin/bash $DIRECTORY/servers/load_balancing_set.sh $balancing_type
        $balancing_server_ip $proxy_address
    else
        echo "Aborted by choosing load-balancing ip..."
    fi
else
    echo "Aborted by choosing load-balancing server..."
fi
```

#### LOAD\_BALANCING\_SET.SH DATEI

---

```
#!/bin/bash
DIRECTORY=/root/scripts/server_proxy
### GET SENT VARIABLES
balancing_type=$1
balancing_server_ip=$2
proxy_address=$3
### LOAD FROM VARIABLES
source $DIRECTORY/variables/balancing_counters
source $DIRECTORY/variables/balancing_addresses
```

```

### SET VARIABLES
case $balancing_type in
    "round")
        config_file_name=${balancing_type}"_$round_count"
        proxy_ip=$round_address
        new_counter=$((round_count+1))
        counter_name=round_count
        ;;
    "weighted")
        config_file_name=${balancing_type}"_$weighted_count"
        proxy_ip=$weighted_address
        new_counter=$((weighted_count+1))
        counter_name=weighted_count
        ;;
    "persistent")
        config_file_name=${balancing_type}"_$persistent_count"
        proxy_ip=$persistent_address
        new_counter=$((persistent_count+1))
        counter_name=persistent_count
        ;;
    "least")
        config_file_name=${balancing_type}"_$least_count"
        proxy_ip=$least_address
        new_counter=$((least_count+1))
        counter_name=least_count
        ;;
esac

echo ""; echo -e "\e[1;31m ADD SERVER TO BALANCING: $balancing_type"; tput sgr0

/bin/bash $DIRECTORY/shared/proxy_balancing_new.sh $balancing_type
$balancing_server_ip

echo ""; echo -e "\e[1;31m UPDATE BALANCING VARIABLE FILE"; tput sgr0
sed -i "$counter_name/ c $counter_name=$new_counter"
$DIRECTORY/variables/balancing_counters

echo ""; echo -e "\e[1;31m CREATE SERVER CONFIG FILE"; tput sgr0
echo "server_name=$config_file_name
type=load-balancing
new_server_ip=$balancing_server_ip
proxy_server_ip=$proxy_ip

```



```
proxy_address=$proxy_address" > $DIRECTORY/added_servers/$config_file_name
```

```
echo "" ; echo -e "\e[1;33m RESTART PROXY"; tput sgr0
service nginx restart
```

#### MENU\_SERVERS.SH DATEI

---

```
#!/bin/bash
DIRECTORY=/root/scripts/server_proxy/servers
while [ option != "0" ]
do
    clear;echo
    echo -e "\e[31mSERVER MENU"; tput sgr0
    echo ""
    echo "1          Show Servers"
    echo "2          Add Server"
    echo "3          Delete Server"
    echo ""
    read -p "Option: " option

    case $option in
        1)
            echo "Showing Servers..."
            /bin/bash $DIRECTORY/show_servers.sh
            break;;
        2)
            echo "Adding Server..."
            /bin/bash $DIRECTORY/new_servers.sh
            break;;
        3)
            echo "Deleting Server..."
            /bin/bash $DIRECTORY/del_servers.sh
            break;;
        *)
            echo "Not a Valid Selection at SERVER MENU"
            break;;
    esac

    echo "Press ENTER to go back to MENU";
done
```

#### NEW\_SERVERS.SH DATEI

---

```
#!/bin/bash
DIRECTORY=/root/scripts/server_proxy/servers
while [ option != "0" ]
do
    clear;echo
    echo -e "\e[31mSELECT SERVER TYPE MENU"; tput sgr0
    echo ""
    echo "1          Add as Single Server"
    echo "2          Add Server to Load-Balancing"
    echo ""
    read -p "Option: " option
    case $option in
        1)
            echo "Adding as Single Server..."
            /bin/bash $DIRECTORY/single_server_new.sh
            break;;
        2)
            echo "Adding Server to Load-Balancing..."
            /bin/bash $DIRECTORY/load_balancing_menu.sh
            break;;
        *)
            echo "Not a Valid Selection at SERVER TYPE MENU"
            break;;
    esac
    echo "Press ENTER to go back to MENU";
done
```

#### SHOW\_SERVERS.SH DATEI

---

```
#!/bin/bash
DIRECTORY=/root/scripts/server_proxy/added_servers
echo ""
if [ "$(ls -A $DIRECTORY)" ]
then
    for entry in "$DIRECTORY"/*
    do
        source $entry
        echo -e "SERVER_NAME: \e[1;33m$server_name"; tput sgr0
        echo -e "          TYPE: \e[1;32m$type"; tput sgr0
        echo "          IP: $new_server_ip"
        echo "          PROXY: $proxy_server_ip"
        if [ $type == single_server ]

```

```

        then
            echo "          ADAPTER: $adapter_name"
        else
            echo "          ADDRESS: $proxy_address"
        fi
    done
else
    echo "No added servers found..."
fi

```

#### SINGLE\_SERVER\_DEL.SH DATEI

---

```

#!/bin/bash
DIRECTORY=/root/scripts/server_proxy
### LOAD FROM FILES
source $DIRECTORY/variables/domain_name
### READ SERVER NAME
read -p "Single Server name to be deleted: " server_name
full_name=${server_name}".$domain_name"
echo "" ; echo -e "Server address: \e[1;31m$full_name"; tput sgr0
read -p "Is that correct(Y/N): " option_server_name

if [ $option_server_name == "Y" ] || [ $option_server_name == "y" ]
then
    entry=$DIRECTORY/added_servers/$server_name

    if [ -f $entry ]
    then
        source $entry

        if [ $type == single_server ]
        then
            echo ""; echo -e "\e[1;31m DELETE NETWORK INTERFACE"; tput sgr0
            /bin/bash $DIRECTORY/shared/interface_del.sh $adapter_name

            echo ""; echo -e "\e[1;31m UPDATE DNS"; tput sgr0
            /bin/bash $DIRECTORY/shared/dns_entry_del.sh $server_name
$new_server_ip $domain_name

            echo "" ; echo -e "\e[1;31m UPDATE PROXY"; tput sgr0
            /bin/bash $DIRECTORY/shared/proxy_single_del.sh $server_name
$full_name

```

```

        echo "" ; echo -e "\e[1;31m UPDATE FIREWALL"; tput sgr0
        /bin/bash $DIRECTORY/shared/fw_single_del.sh $server_name
$new_server_ip

        echo "" ; echo -e "\e[1;31m DELETE SERVER CONFIG FILE"; tput sgr0
        rm -rfv $DIRECTORY/added_servers/$server_name
    else
        echo "Server chosen is not a Single Server..."
    fi
else
    echo "server_name not found :("
fi
else
    echo "Aborted by choosing single server name..."
fi

```

#### SINGLE\_SERVER\_NEW.SH DATEI

```

#!/bin/bash
DIRECTORY=/root/scripts/server_proxy
### LOAD FROM FILES
source $DIRECTORY/variables/domain_name
### READ SERVER NAME
read -p "Single Server name to be added: " server_name
full_name=${server_name}".$domain_name"
echo -e "New server name: \e[1;31m$full_name"; tput sgr0
read -p "Is that correct(Y/N): " option_server_name

if [ $option_server_name == "Y" ] || [ $option_server_name == "y" ]
then
    read -p "Please enter the IP adress: " proxy_server_ip
    echo -e "New server IP: \e[1;31m$proxy_server_ip"; tput sgr0
    read -p "Is that correct(Y/N): " option_server_ip

    if [ $option_server_ip == "Y" ] || [ $option_server_ip == "y" ]
    then
        /bin/bash $DIRECTORY/servers/single_server_set.sh $server_name
$proxy_server_ip $domain_name $full_name
    else
        echo "Aborted by choosing single server ip..."
    fi
fi

```

```

else
    echo "Aborted by choosing single server name..."
fi

```

#### SINGLE\_SERVER\_SET.SH DATEI

---

```

#!/bin/bash
DIRECTORY=/root/scripts/server_proxy
### GET SENT VARIABLES
server_name=$1
proxy_server_ip=$2
domain_name=$3
full_name=$4
### LOAD FROM FILES
source $DIRECTORY/variables/new_ip
source $DIRECTORY/variables/adapter_number
### SET VARIABLES
new_server_ip=172.16.2.$server_ip
adapter_name=eth1:added$adapter_number

echo "" ; echo -e "\e[1;31m NEW NETWORK INTERFACE"; tput sgr0
/bin/bash $DIRECTORY/shared/interface_new.sh $new_server_ip $adapter_name

echo "" ; echo -e "\e[1;31m UPDATE DNS"; tput sgr0
/bin/bash $DIRECTORY/shared/dns_entry_new.sh $server_name $new_server_ip
$server_ip $domain_name $full_name

echo "" ; echo -e "\e[1;31m UPDATE PROXY"; tput sgr0
/bin/bash $DIRECTORY/shared/proxy_single_new.sh $server_name $new_server_ip
$proxy_server_ip $full_name

echo "" ; echo -e "\e[1;31m UPDATE FIREWALL"; tput sgr0
/bin/bash $DIRECTORY/shared/fw_single_new.sh $server_name $new_server_ip

echo ""; echo -e "\e[1;31m CREATE SERVER CONFIG FILE"; tput sgr0
echo "server_name=$server_name
type=single_server
new_server_ip=$proxy_server_ip
proxy_server_ip=$new_server_ip
adapter_name=$adapter_name" > $DIRECTORY/added_servers/$server_name

```

---

#### SHARED ORDNER

#### DNS\_ENTRY\_DEL.SH DATEI

---

```
#!/bin/bash
### GET SENT VARIABLES
server_name=$1
server_ip=$2
domain_name=$3
forward_name="db.${domain_name}.forward"
rev_name="db.${domain_name}.rev"

echo "" ; echo -e "\e[1;33m REMOVING FROM: /etc/bind/$forward_name"; tput sgr0
sed -i "/$server_name IN A $server_ip/d" /etc/bind/$forward_name

echo -e "\e[1;33m REMOVING FROM: /etc/bind/$rev_name"; tput sgr0
### STRIP
ip_server="${new_server_ip//172.16.2.}"
### UPDATE
sed -i "/$ip_server IN PTR $server_name.$domain_name/d" /etc/bind/$rev_name

echo -e "\e[1;32m RESTARTING DNS..."; tput sgr0
service bind9 restart
```

---

#### DNS\_ENTRY\_NEW.SH DATEI

```
#!/bin/bash
### GET SENT VARIABLES
server_name=$1
new_server_ip=$2
server_ip=$3
domain_name=$4
full_name=$5
forward_name="db.${domain_name}.forward"
rev_name="db.${domain_name}.rev"

echo "" ; echo -e "\e[1;33m ADDING TO: /etc/bind/$forward_name"; tput sgr0
echo "$server_name IN A $new_server_ip" >> /etc/bind/$forward_name

echo -e "\e[1;33m ADDING TO: /etc/bind/$rev_name"; tput sgr0
echo "$server_ip IN PTR $full_name" >> /etc/bind/$rev_name

echo -e "\e[1;32m RESTARTING DNS..."; tput sgr0
service bind9 restart
```

---

#### FW\_SINGLE\_DEL.SH DATEI

```
#!/bin/bash
DIRECTORY=/root/scripts/firewall
### GET SENT VARIABLES
server_name=$1
new_server_ip=$2
server_name_ip=${server_name}_ip"

echo "" ; echo -e "\e[1;33m REMOVING FROM:
/firewall/servers/servers_singles: $server_name_ip=$new_server_ip"; tput
sgr0

sed -i "/$server_name_ip=$new_server_ip/d"
$DIRECTORY/servers/servers_singles

echo -e "\e[1;33m REMOVING RULES FROM: /firewall/7_servers_singles.sh";
tput sgr0

sed -i "/### BEGIN: $server_name/,/### END: $server_name/d"
$DIRECTORY/7_servers_singles.sh

echo -e "\e[1;32m RESTARTING FIREWALL"; tput sgr0
/bin/bash /root/scripts/fw_up.sh
```

#### FW\_SINGLE\_NEW.SH DATEI

---

```
#!/bin/bash
DIRECTORY=/root/scripts/firewall
### GET SENT VARIABLES
server_name=$1
new_server_ip=$2

echo ""
echo -e "\e[1;33m ADDING TO /firewall/servers/servers_singles:
$server_name=$new_server_ip"; tput sgr0
server_name_ip=${server_name}_ip"
echo -e "$server_name_ip=$new_server_ip" >>
$DIRECTORY/servers/servers_singles

echo -e "\e[1;33m ADDING RULES TO /firewall/7_servers_singles.sh"; tput
sgr0
echo "### BEGIN: $server_name" >> $DIRECTORY/7_servers_singles.sh

echo "\$fw -A INPUT -p icmp --icmp-type 8 -i \$to_server_adapter -s \$lanB
-d \$server_name_ip -j ACCEPT" >> $DIRECTORY/7_servers_singles.sh
echo "\$fw -A INPUT -p icmp --icmp-type 8 -i \$to_server_adapter -s \$lanC
-d \$server_name_ip -j ACCEPT" >> $DIRECTORY/7_servers_singles.sh
```

```
echo "\$fw -A INPUT -p tcp --dport \$port_http -i \$to_server_adapter -s  
\$lanB -d \$server_name_ip -j ACCEPT" >> $DIRECTORY/7_servers_singles.sh  
echo "\$fw -A INPUT -p tcp --dport \$port_http -i \$to_server_adapter -s  
\$lanC -d \$server_name_ip -j ACCEPT" >> $DIRECTORY/7_servers_singles.sh
```

```
echo "### END: $server_name" >> $DIRECTORY/7_servers_singles.sh
```

```
echo -e "\e[1;32m RESTARTING FIREWALL"; tput sgr0  
/bin/bash /root/scripts/fw_up.sh
```

#### INTERFACE\_DEL.SH DATEI

---

```
#!/bin/bash  
### GET SENT VARIABLES  
adapter_name=$1  
  
echo "" ; echo -e "\e[1;33m BRING INTERFACE DOWN: $adapter_name"; tput sgr0  
ifdown $adapter_name
```

```
echo -e "\e[1;33m UPDATE: /etc/network/interfaces"; tput sgr0  
sed -i "/auto $adapter_name/,/255.255.255.0/d" /etc/network/interfaces
```

```
echo -e "\e[1;32m RESTART NETWORKING"; tput sgr0  
service networking restart
```

#### INTERFACE\_NEW.SH DATEI

---

```
#!/bin/bash  
DIRECTORY=/root/scripts/server_proxy  
### GET SENT VARIABLES  
new_server_ip=$1  
adapter_name=$2  
### LOAD FROM FILES  
source $DIRECTORY/variables/new_ip  
source $DIRECTORY/variables/adapter_number  
  
echo ""  
echo -e "\e[1;33m UPDATING single_ip IN: variables/single_ip"; tput sgr0  
new_single_ip=$((server_ip+1))  
echo server_ip=$new_single_ip > $DIRECTORY/variables/new_ip  
  
echo -e "\e[1;33m UPDATING adapter_number IN: variables/adapter_number";  
tput sgr0  
new_adapter_number=$((adapter_number+1))
```



```

echo adapter_number=$new_adapter_number >
$DIRECTORY/variables/adapter_number

echo -e "\e[1;33m ADDING NEW INTERFACE IN: /etc/network/interfaces"; tput
sgr0

echo "auto $adapter_name
iface $adapter_name inet static
    address $new_server_ip
    netmask 255.255.255.0" >> /etc/network/interfaces

echo -e "\e[1;32m BRING UP NEW ADAPTER"; tput sgr0
ifup $adapter_name

echo -e "\e[1;32m RESTART NETWORKING"; tput sgr0
service networking restart

```

#### PROXY\_BALANCING\_NEW.SH DATEI

---

```

#!/bin/bash
### GET SENT VARIABLES
balancing_type=$1
balancing_server_ip=$2
proxy_server_port=$3

if [ -z "$proxy_server_port" ]
then
    proxy_server_port=80
fi

if [ $balancing_type == weighted ] || [ $balancing_type == c-weighted ]
then
    read -p "Want to add weight-value to the server?(Y/N): " option_weight
    if [ $option_weight == "Y" ] || [ $option_weight == "y" ]
    then
        read -p "Enter weight-value: " weight_value
        echo -e "New weight-value: \e[1;31m$weight_value"; tput sgr0
        read -p "Is that correct(Y/N): " option_confirm_weight
        if [ $option_confirm_weight == "Y" ] || [ $option_confirm_weight == "y" ]
        ]
        then
            sed -i "/TO ADD/{s/.*/&\n    server
$balancing_server_ip:$proxy_server_port weight=$weight_value;/:a;n;ba}"
/etc/nginx/sites-available/$balancing_type

```

```

        else
            break
        fi
    else
        echo "Adding without weight-value..."
        sed -i "/TO ADD/{s/.*&\n    server
$balancing_server_ip:$proxy_server_port;/:a;n;ba}" /etc/nginx/sites-
available/$balancing_type
        fi
    else
        sed -i "/TO ADD/{s/.*&\n    server
$balancing_server_ip:$proxy_server_port;/:a;n;ba}" /etc/nginx/sites-
available/$balancing_type
        fi
    fi

```

---

#### PROXY\_SINGLE\_DEL.SH DATEI

```

#!/bin/bash

### GET SENT VARIABLES
server_name=$1
full_name=$2

echo "" ; echo -e "\e[1;33m DISABLE PROXY: $full_name"; tput sgr0
rm -rfv /etc/nginx/sites-enabled/$server_name

echo -e "\e[1;33m REMOVING: /etc/nginx/sites-available/$server_name"; tput
sgr0
rm -rfv /etc/nginx/sites-available/$server_name

echo -e "\e[1;32m RESTARTING PROXY..."; tput sgr0
service nginx restart

```

---

#### PROXY\_SINGLE\_NEW.SH DATEI

```

#!/bin/bash

DIRECTORY=/root/scripts/server_proxy

### GET SENT VARIABLES
server_name=$1
new_server_ip=$2
proxy_server_ip=$3
full_name=$4
proxy_server_port=$5

### CHECK IF PORT NUMBER WAS SENT: THIS WORKS FOR SINGLE AND CONTAINER
if [ -z "$proxy_server_port" ]

```

```

then
    proxy_server_port=80
fi
echo "" ; echo -e "\e[1;33m ADDING TO /etc/nginx/sites-
available/$server_name"; tput sgr0
echo "server {
    listen $new_server_ip:80;
    server_name $full_name;

    location / {
        proxy_pass http://$proxy_server_ip:$proxy_server_port;
        proxy_redirect default;
        proxy_read_timeout 900;
        proxy_send_timeout 900;

        ### HTACCESS: Enable User Login
        auth_basic 'Restricted';
        auth_basic_user_file /etc/nginx/htaccess/.htpasswd;
    }
}" > /etc/nginx/sites-available/$server_name

echo -e "\e[1;33m ENABLE PROXY: $full_name"; tput sgr0
ln -s /etc/nginx/sites-available/$server_name /etc/nginx/sites-
enabled/$server_name

echo -e "\e[1;32m RESTARTING PROXY..."; tput sgr0
service nginx restart

```

---

## USERS ORDNER

### DEL\_USER.SH DATEI

---

```

#!/bin/bash

echo "" ; echo -e "\e[31mDELETE USER"; tput sgr0
read -p "Please user name: " del_user_name
echo -e "User name: \e[1;31m$del_user_name"; tput sgr0
read -p "Is that correct(Y/N): " option_user_name
if [ $option_user_name == "Y" ] || [ $option_user_name == "y" ]
then
    file=/etc/nginx/htaccess/.htpasswd
    search=`grep $del_user_name $file`
    if [ $search ]

```

```

then
    htpasswd -D $file $del_user_name
else
    echo "User to be deleted not found..."
fi
else
    echo "Aborted by choosing user name..."
fi

```

## MENU\_USERS.SH DATEI

---

```

#!/bin/bash
DIRECTORY=/root/scripts/server_proxy/users

while [ option != "0" ]
do
    clear;echo
    echo -e "\e[31mUSERS MENU"; tput sgr0
    echo ""
    echo "1          Show Users"
    echo "2          Add User"
    echo "3          Update User"
    echo "4          Delete User"
    echo ""
    read -p "Option: " option
    case $option in
        1)
            echo "Show Users..."
            /bin/bash $DIRECTORY/show_users.sh
            break;;
        2)
            echo "Adding User..."
            /bin/bash $DIRECTORY/new_users.sh
            break;;
        3)
            echo "Updating User..."
            /bin/bash $DIRECTORY/update_users.sh
            break;;
        4)
            echo "Deleting User..."
            /bin/bash $DIRECTORY/del_users.sh
            break;;
    esac
done

```

```

        *)
            echo "Not a Valid Selection at USERS Menu"
            break;;
        esac
        echo "Press ENTER to go back to MENU";
done

```

#### NEW\_USERS.SH DATEI

---

```

#!/bin/bash

echo "" ; echo -e "\e[31mADD NEW USER"; tput sgr0
read -p "Please enter new user name: " new_user_name
echo -e "New user name: \e[1;31m$new_user_name"; tput sgr0
read -p "Is that correct(Y/N): " option_user_name
if [ $option_user_name == "Y" ] || [ $option_user_name == "y" ]
then
    htpasswd /etc/nginx/htaccess/.htpasswd $new_user_name
else
    echo "Aborted by choosing new user name..."
fi

```

#### SHOW\_USERS.SH DATEI

---

```

#!/bin/bash

file=/etc/nginx/htaccess/.htpasswd

echo ""
while read -r line
do
    user=${line%:*}
    echo -e "\e[1;33m $user"; tput sgr0
done < $file
echo ""

```

#### UPDATE\_USERS.SH DATEI

---

```

#!/bin/bash

echo "" ; echo -e "\e[31mUPDATE USER"; tput sgr0
read -p "Please user name: " user_name
echo -e "User name: \e[1;31m$user_name"; tput sgr0
read -p "Is that correct(Y/N): " option_user_name
if [ $option_user_name == "Y" ] || [ $option_user_name == "y" ]
then

```

```

file=/etc/nginx/.htpasswd
search=`grep $user_name $file`
if [ $search ]
then
    htpasswd $file $user_name
else
    echo "User to be updated not found..."
fi
else
    echo "Aborted by choosing user name..."
fi

```

---

## VARIABLES ORDNER

### ADAPTER\_NUMBER DATEI

---

Hier wird es gespeichert und aktualisiert nach der wahl von eingesetzten und neuen Netzwerkschnittstellen zum Server.

```

root@proxy:~# cat ~/scripts/server_proxy/variables/adapter_number
adapter_number=1

```

### BALANCING\_ADDRESSES DATEI

---

Hier sind die Load-Balancing Adressen vom System

```

root@proxy:~# cat ~/scripts/server_proxy/variables/balancing_addresses
round_address=172.16.2.12
weighted_address=172.16.2.13
persistent_address=172.16.2.14
least_address=172.16.2.15
c_round_address=172.16.2.22
c_weighted_address=172.16.2.23
c_persistent_address=172.16.2.24
c_least_address=172.16.2.25

```

### BALANCING\_COUNTERS DATEI

---

Der Zähler des eingesetzten Servers im System.

```

root@proxy:~# cat ~/scripts/server_proxy/variables/balancing_counters
round_count=1
weighted_count=1
persistent_count=1
least_count=1
round_count=1

```

```
weighted_count=1
c_persistent_count=1
c_least_count=1
```

## DOMAIN\_NAME DATEI

---

Hier wird der Domainname Name eingesetzt.

```
root@proxy:~# cat ~/scripts/server_proxy/variables/domain_name
domain_name=webproxy.local
```

## NEW\_IP DATEI

---

Hier wird die IP-Adresse der Server gespeichert.Nachdem dieser Aktualisiert wurde.

```
root@proxy:~# cat ~/scripts/server_proxy/variables/new_ip
server_ip=100
```

## TESTS ANLAGEN

### TEST ANLAGE 1: CLIENTS IP-ADRESSE TESTS

---

#### CLIENT 1 IP-ADRESSE TESTS

---

##### HOSTNAME -I

---

```
web@cli1 ~ $ hostname -I
192.168.3.11
```

##### DNSDOMAINNAME

---

```
web@cli1 ~ $ dnsdomainname
name.local
```

##### IP ADDR SHOW

---

```
web@cli1 ~ $ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast
state DOWN group default qlen 1000
    link/ether 08:00:27:80:d7:16 brd ff:ff:ff:ff:ff:ff
```

```

    inet 169.254.6.153/16 brd 169.254.255.255 scope link eth0:avahi
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP group default qlen 1000
    link/ether 08:00:27:36:41:21 brd ff:ff:ff:ff:ff:ff
    inet 192.168.3.11/24 brd 192.168.3.255 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe36:4121/64 scope link
        valid_lft forever preferred_lft forever

```

---

## CLIENT 2 IP-ADRESSE TESTS

---

### HOSTNAME -I

```
web@cli2 ~ $ hostname -I
```

```
web@cli2 ~ $
```

---

### DNSDOMAINNAME

```
web@cli2 ~ $ dnsdomainname
```

```
name.local
```

```
web@cli2 ~ $
```

---

### IP ADDR SHOW

```
web@cli2:~$ ip addr show
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default
```

```
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
```

```
    inet 127.0.0.1/8 scope host lo
```

```
        valid_lft forever preferred_lft forever
```

```
    inet6 ::1/128 scope host
```

```
        valid_lft forever preferred_lft forever
```

```
2: enp0s3: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast
state DOWN group default qlen 1000
```

```
    link/ether 08:00:27:34:3f:d2 brd ff:ff:ff:ff:ff:ff
```

```
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP group default qlen 1000
```

```
    link/ether 08:00:27:10:d9:08 brd ff:ff:ff:ff:ff:ff
```

```
    inet6 fe80::a00:27ff:fe10:d908/64 scope link
```

```
        valid_lft forever preferred_lft forever
```

---

## CLIENT 3 IP-ADRESSE TESTS

---

### IPCONFIG

```
C:\Users\Web>ipconfig /all
```



#### Windows-IP-Konfiguration

```
Hostname . . . . . : cli3
Prim„res DNS-Suffix . . . . . :
Knotentyp . . . . . : Hybrid
IP-Routing aktiviert . . . . . : Nein
WINS-Proxy aktiviert . . . . . : Nein
DNS-Suffixsuchliste . . . . . : name.local
```

#### Ethernet-Adapter LAN-Verbindung 2:

```
Verbindungsspezifisches DNS-Suffix: name.local
Beschreibung. . . . . : Intel(R) PRO/1000 MT-Desktopadapter
#2
Physikalische Adresse . . . . . : 08-00-27-A9-90-D7
DHCP aktiviert. . . . . : Ja
Autokonfiguration aktiviert . . . : Ja
Verbindungslokale IPv6-Adresse . :
fe80::d5:7740:c692:7fb5%14 (Bevorzugt)
IPv4-Adresse . . . . . : 192.168.3.33 (Bevorzugt)
Subnetzmaske . . . . . : 255.255.255.0
Lease erhalten. . . . . : Sonntag, 8. Mai 2016 14:56:45
Lease l„uft ab. . . . . : Sonntag, 8. Mai 2016 15:11:45
Standardgateway . . . . . : 192.168.3.1
                             172.16.2.1
DHCP-Server . . . . . : 192.168.3.1
DHCPv6-IAID . . . . . : 319291431
DHCPv6-Client-DUID. . . . . : 00-01-00-01-1C-25-C6-6B-08-00-27-D8-
91-8C
DNS-Server . . . . . : 192.168.3.1
NetBIOS Über TCP/IP . . . . . : Aktiviert
```

#### TEST ANLAGE 2: NAMEN AUFLÖSUNG VON IMITATION SERVER TESTS

##### CLIENT 1 NAMEN AUFLÖSUNG VON IMITATION SERVER TESTS

###### NSLOOKUP

```
web@cli1 ~ $ nslookup dns.name.local
Server:          192.168.3.1
Address:         192.168.3.1#53
```

```
Name: dns.name.local
Address: 192.168.3.1
```

---

## DIG

```
web@cli1 ~ $ dig dns.name.local
```

```
; <<>> DiG 9.9.5-3ubuntu0.5-Ubuntu <<>> dns.name.local
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 64433
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;dns.name.local.                IN      A

;; ANSWER SECTION:
dns.name.local.                86400 IN      A      192.168.3.1

;; AUTHORITY SECTION:
name.local.                    86400 IN      NS      dns.name.local.

;; Query time: 1 msec
;; SERVER: 192.168.3.1#53(192.168.3.1)
;; WHEN: Thu May 12 16:56:51 CEST 2016
;; MSG SIZE rcvd: 73
```

---

## WHOIS

```
web@cli1 ~ $ whois -H proxy.webproxy.local
```

Für diese Art von Objekten ist kein Whois-Server bekannt.

---

## CLIENT 2 NAMEN AUFLÖSUNG VON IMITATION SERVER TESTS

---

### NSLOOKUP

```
web@cli2:~$ nslookup dns.name.local
```

```
; connection timed out; no servers could be reached
```

---

### DIG

```
web@cli2:~$ dig dns.name.local
```

```
; <<>> DiG 9.9.5-11ubuntu1-Ubuntu <<>> dns.name.local
;; global options: +cmd
;; connection timed out; no servers could be reached
```

---

### WHOIS

```
web@cli1 ~ $ whois -H proxy.webproxy.local
```

Für diese Art von Objekten ist kein Whois-Server bekannt.

---

## CLIENT 3 NAMEN AUFLÖSUNG VON IMITATION SERVER TESTS

---

### NSLOOKUP

```
C:\Users\web>nslookup dns.name.local
```

Server: Unknown

Address: 192.168.3.1

Name: dns.name.local

Address: 192.168.3.1

## TEST ANLAGE 3: CLIENTS NAMEN AUFLÖSUNG VON PROXY SERVER TESTS

---

## CLIENT 1 NAMEN AUFLÖSUNG VON PROXY SERVER TESTS

---

### NSLOOKUP

```
web@cli1 ~ $ nslookup proxy.webproxy.local
```

Server: 192.168.3.1

Address: 192.168.3.1#53

Non-authoritative answer:

Name: proxy.webproxy.local

Address: 172.16.2.2

---

### DIG

```
web@cli1 ~ $ dig proxy.webproxy.local
```

```
; <<>> DiG 9.9.5-3ubuntu0.5-Ubuntu <<>> proxy.webproxy.local
```

```
;; global options: +cmd
```

```
;; Got answer:
```

```
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 23864
```

```
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 3
```

```
;; OPT PSEUDOSECTION:
```

```
; EDNS: version: 0, flags:; udp: 4096
```

```
;; QUESTION SECTION:
```

```
;proxy.webproxy.local. IN A
```

```
;; ANSWER SECTION:
```

```
proxy.webproxy.local. 604775 IN A 172.16.2.2
```

```
;; AUTHORITY SECTION:
```

```
webproxy.local.      604775      IN      NS      webproxy.local.
```

```
;; ADDITIONAL SECTION:
```

```
webproxy.local.      604775      IN      A       172.16.2.2
```

```
webproxy.local.      604775      IN      AAAA    ::1
```

```
;; Query time: 1 msec
```

```
;; SERVER: 192.168.3.1#53(192.168.3.1)
```

```
;; WHEN: Thu May 12 17:05:54 CEST 2016
```

```
;; MSG SIZE  rcvd: 123
```

---

#### CLIENT 2 NAMEN AUFLÖSUNG VON PROXY SERVER TESTS

---

##### NSLOOKUP

```
web@cli2 ~ $ nslookup proxy.webproxy.local
```

```
;; connection timed out; no servers could be reached
```

---

##### DIG

```
web@cli2 ~ $ dig proxy.webproxy.local
```

```
; <<>> DiG 9.9.5-11ubuntu1-Ubuntu <<>> dns.name.local
```

```
;; global options: +cmd
```

```
;; connection timed out; no servers could be reached
```

---

#### CLIENT 3 NAMEN AUFLÖSUNG VON PROXY SERVER TESTS

---

##### NSLOOKUP

```
C:\Users\web>nslookup proxy.name.local
```

```
Server: Unknown
```

```
Address: 192.168.3.1
```

```
Name: proxy.webproxy.local
```

```
Address: 172.16.2.2
```

---

#### TEST ANLAGE 4: CLIENTS NAMEN AUFLÖSUNG VON TESTUMGEBUNG TESTS

---

#### CLIENT 1 NAMEN AUFLÖSUNG VON TESTUMGEBUNG TESTS

---

##### NSLOOKUP

```
web@cli1 ~ $ nslookup single.webproxy.local
```

```
Server:          192.168.3.1
```

```
Address:        192.168.3.1#53
```

```
Non-authoritative answer:
```

```
Name: single.webproxy.local
```

Address: 172.16.2.11

---

#### DIG

```
web@cli1 ~ $ dig single.webproxy.local
; <<>> DiG 9.9.5-3ubuntu0.5-Ubuntu <<>> single.webproxy.local
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 50533
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;single.webproxy.local.      IN      A

;; ANSWER SECTION:
single.webproxy.local. 604753      IN      A      172.16.2.11

;; AUTHORITY SECTION:
webproxy.local.        604753      IN      NS      webproxy.local.

;; ADDITIONAL SECTION:
webproxy.local.        604753      IN      A      172.16.2.2
webproxy.local.        604753      IN      AAAA    ::1

;; Query time: 29 msec
;; SERVER: 192.168.3.1#53(192.168.3.1)
;; WHEN: Sat May 14 19:49:22 CEST 2016
;; MSG SIZE rcvd: 124
```

---

#### CLIENT 2 NAMEN AUFLÖSUNG VON TESTUMGEBUNG TESTS

---

#### DIG

```
web@cli2 ~ $ dig single.webproxy.local
; <<>> DiG 9.9.5-11ubuntu1-Ubuntu <<>> dns.name.local
;; global options: +cmd
;; connection timed out; no servers could be reached
```

---

#### NSLOOKUP

```
web@cli2 ~ $ nslookup single.webproxy.local
;; connection timed out; no servers could be reached
```

---

## CLIENT 3 NAMEN AUFLÖSUNG VON TESTUMGEBUNG TESTS

---

### NSLOOKUP

```
C:\Users\web>nslookup single.webproxy.local
```

```
Server: UnKnown
```

```
Address: 192.168.3.1
```

```
Name: single.webproxy.local
```

```
Address: 172.16.2.11
```

---

## TEST ANLAGE 5: CLIENTS VERBINDUNG TESTS

---

### VERBINDUNG ZU IMITATION SERVER TESTS

---

#### TRACEROUTE VON CLIENT 1

```
web@cli1 ~ $ traceroute 192.168.3.1
```

```
traceroute to 192.168.3.1 (192.168.3.1), 30 hops max, 60 byte packets
```

```
1 192.168.3.1 (192.168.3.1) 0.909 ms 0.692 ms 0.586 ms
```

---

#### TRACEROUTE VON CLIENT 2

```
web@cli2:~$ traceroute 192.168.3.1
```

```
traceroute to 192.168.3.1 (192.168.3.1), 30 hops max, 60 byte packets
```

```
connect: Das Netzwerk ist nicht erreichbar
```

---

#### TRACEROUTE VON CLIENT 3

```
C:\Users\web>tracert dns.name.local
```

```
Routenverfolgung zu dns.name.local [192.168.3.1] über maximal 30
```

```
Abschnitte:
```

```
1 <1 ms <1 ms <1 ms 192.168.3.1
```

```
Ablaufverfolgung beendet.
```

---

### VERBINDUNG ZU PROXY SERVER TESTS

---

#### TRACEROUTE VON CLIENT 1

```
web@cli1 ~ $ traceroute 172.16.2.2
```

```
traceroute to 172.16.2.2 (172.16.2.2), 30 hops max, 60 byte packets
```

```
1 192.168.3.1 (192.168.3.1) 0.281 ms 0.197 ms 0.158 ms
```

```
2 172.16.2.2 (172.16.2.2) 0.449 ms 0.415 ms 0.295 ms
```

---

#### TRACEROUTE VON CLIENT 2

```
web@cli2:~$ traceroute 172.16.2.2
```

```
traceroute to 172.16.2.2 (172.16.2.2), 30 hops max, 60 byte packets
```

```
connect: Das Netzwerk ist nicht erreichbar
```

---

### TRACEROUTE VON CLIENT 3

```
C:\Users\web>tracert dns.name.local
```

```
Routenverfolgung zu proxy.webproxy.local [172.16.2.2] über maximal 30
Abschnitte:
```

```
 1    <1 ms    <1 ms    <1 ms    192.168.3.1
 2    <1 ms    <1 ms    <1 ms    172.16.2.2
```

```
Ablaufverfolgung beendet.
```

---

### VERBINDUNG ZU TESTUMGEBUNG TESTS

---

#### TRACEROUTE VON CLIENT 1

```
web@cli1 ~ $ traceroute 172.16.2.11
```

```
traceroute to 172.16.2.11 (172.16.2.11), 30 hops max, 60 byte packets
```

```
 1  192.168.3.1 (192.168.3.1)  0.227 ms  0.148 ms  0.108 ms
 2  172.16.2.11 (172.16.2.11)  0.403 ms  0.304 ms  0.302 ms
```

---

#### TRACEROUTE VON CLIENT 2

```
web@cli2:~$ traceroute 172.16.2.11
```

```
traceroute to 172.16.2.2 (172.16.2.2), 30 hops max, 60 byte packets
```

```
connect: Das Netzwerk ist nicht erreichbar
```

---

#### TRACEROUTE VON CLIENT 3

```
C:\Users\web>tracert single.webproxy.local
```

```
Routenverfolgung zu single.webproxy.local [172.16.2.11] über maximal 30
Abschnitte:
```

```
 1    <1 ms    <1 ms    <1 ms    192.168.3.1
 2    <1 ms    <1 ms    <1 ms    172.16.2.11
```

```
Ablaufverfolgung beendet.
```

---

### TEST ANLAGE 6: SICHERHEITSTESTS

---

#### PROXY SERVER NETZWERDIENST

```
root@proxy:~# netstat -plunt
```

```
Aktive Internetverbindungen (Nur Server)
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program
tcp	0	0	172.16.2.13:80	0.0.0.0:*	LISTEN	1712/nginx
tcp	0	0	172.16.2.10:80	0.0.0.0:*	LISTEN	1712/nginx
tcp	0	0	172.16.2.11:80	0.0.0.0:*	LISTEN	1712/nginx
tcp	0	0	172.16.2.12:80	0.0.0.0:*	LISTEN	1712/nginx
tcp	0	0	172.16.2.14:80	0.0.0.0:*	LISTEN	1712/nginx
tcp	0	0	172.16.2.15:80	0.0.0.0:*	LISTEN	1712/nginx
tcp	0	0	172.16.2.23:80	0.0.0.0:*	LISTEN	1712/nginx
tcp	0	0	172.16.2.21:80	0.0.0.0:*	LISTEN	1712/nginx

tcp	0	0	172.16.2.22:80	0.0.0.0:*	LISTEN	1712/nginx
tcp	0	0	172.16.2.24:80	0.0.0.0:*	LISTEN	1712/nginx
tcp	0	0	172.16.2.25:80	0.0.0.0:*	LISTEN	1712/nginx
tcp	0	0	172.16.2.2:53	0.0.0.0:*	LISTEN	1676/named
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	1636/sshd
tcp	0	0	127.0.0.1:953	0.0.0.0:*	LISTEN	1676/named
tcp6	0	0	:::53	:::*	LISTEN	1676/named
tcp6	0	0	:::22	:::*	LISTEN	1636/sshd
udp	0	0	0.0.0.0:47401	0.0.0.0:*		614/dhclient
udp	0	0	172.16.2.2:53	0.0.0.0:*		1676/named
udp	0	0	0.0.0.0:68	0.0.0.0:*		614/dhclient
udp6	0	0	:::53	:::*		1676/named
udp6	0	0	:::7001	:::*		614/dhclient

---

## PROXY SERVER SICHERHEITSTESTS

---

### OHNE FIREWALL

---

#### NMAP AGGRESSIVE MODE OHNE FIREWALL

---

```
web@clil1:~$ sudo nmap -sV -T4 -A proxy.webproxy.local
```

```
Starting Nmap 6.47 ( http://nmap.org ) at 2016-05-15 17:14 CEST
```

```
Nmap scan report for proxy.webproxy.local (172.16.2.2)
```

```
Host is up (0.00055s latency).
```

```
Not shown: 998 closed ports
```

```
PORT      STATE SERVICE VERSION
```

```
22/tcp open  ssh      (protocol 2.0)
```

```
| ssh-hostkey:
```

```
| 1024 4e:71:82:f8:f4:e0:da:d3:6c:9d:80:4c:03:1b:b3:84 (DSA)
```

```
| 2048 09:53:9b:ac:d1:a2:aa:4e:cd:7a:6e:a0:94:28:3b:d0 (RSA)
```

```
|_ 256 c5:ab:7c:38:40:d9:b2:0a:59:50:c5:b2:49:81:9b:4e (ECDSA)
```

```
53/tcp open  domain
```

```
1 service unrecognized despite returning data. If you know the
service/version, please submit the following fingerprint at
http://www.insecure.org/cgi-bin/servicefp-submit.cgi :
```

```
SF-Port22-TCP:V=6.47%I=7%D=5/15%Time=57389254%P=x86_64-pc-linux-gnu%r(NULL
SF:.,2B,"SSH-2\0-OpenSSH_6\0.6\0.1p1\0x20Ubuntu-2ubuntu2\0.6\r\n");
```

```
Device type: general purpose
```

```
Running: Linux 3.X
```

```
OS CPE: cpe:/o:linux:linux_kernel:3
```

```
OS details: Linux 3.11 - 3.14
```

```
Network Distance: 2 hops
```



TRACEROUTE (using port 995/tcp)

HOP	RTT	ADDRESS
1	0.14 ms	<b>192.168.3.1</b>
2	1.27 ms	<b>172.16.2.2</b>

OS and Service detection performed. Please report any incorrect results at <http://nmap.org/submit/> .

Nmap done: 1 IP address (1 host up) scanned in 18.70 seconds

#### UDP OHNE FIREWALL

---

```
web@clil1:~$ sudo nmap -sU --open --reason --max-retries 1 proxy.webproxy.local
```

```
Starting Nmap 6.47 ( http://nmap.org ) at 2016-05-15 18:47 CEST
Warning: 172.16.2.2 giving up on port because retransmission cap hit (1).
Nmap scan report for proxy.webproxy.local (172.16.2.2)
Host is up, received echo-reply (0.00042s latency).
Not shown: 907 open|filtered ports, 92 closed ports
Reason: 907 no-responses and 92 port-unreaches

PORT      STATE SERVICE REASON
53/udp open  domain  udp-response

Nmap done: 1 IP address (1 host up) scanned in 110.27 seconds
```

#### TCP SYN OHNE FIREWALL

---

```
web@clil1:~$ sudo nmap -sS -Pn proxy.webproxy.local

Starting Nmap 6.47 ( http://nmap.org ) at 2016-05-15 17:16 CEST
Nmap scan report for proxy.webproxy.local (172.16.2.2)
Host is up (0.00032s latency).
Not shown: 998 closed ports

PORT      STATE SERVICE
22/tcp open  ssh
53/tcp open  domain

Nmap done: 1 IP address (1 host up) scanned in 1.63 seconds
```

#### ACK OHNE FIREWALL

---

```
web@clil1:~$ sudo nmap -sA -Pn proxy.webproxy.local

Starting Nmap 6.47 ( http://nmap.org ) at 2016-05-15 17:17 CEST
Nmap scan report for proxy.webproxy.local (172.16.2.2)
Host is up (0.00021s latency).
All 1000 scanned ports on proxy.webproxy.local (172.16.2.2) are unfiltered
Nmap done: 1 IP address (1 host up) scanned in 1.63 seconds
```

#### CONNECT OHNE FIREWALL

---

```
web@cli1:~$ sudo nmap -sT -Pn proxy.webproxy.local
Starting Nmap 6.47 ( http://nmap.org ) at 2016-05-15 17:18 CEST
Nmap scan report for proxy.webproxy.local (172.16.2.2)
Host is up (0.0026s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
53/tcp    open  domain
Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
```

#### WINDOW OHNE FIREWALL

---

```
web@cli1:~$ sudo nmap -sW -Pn proxy.webproxy.local
Starting Nmap 6.47 ( http://nmap.org ) at 2016-05-15 17:19 CEST
Nmap scan report for proxy.webproxy.local (172.16.2.2)
Host is up (0.00022s latency).
All 1000 scanned ports on proxy.webproxy.local (172.16.2.2) are closed
Nmap done: 1 IP address (1 host up) scanned in 1.63 seconds
```

#### MAIMON SCANS OHNE FIREWALL

---

```
web@cli1:~$ sudo nmap -sM -Pn proxy.webproxy.local
Starting Nmap 6.47 ( http://nmap.org ) at 2016-05-15 17:20 CEST
Nmap scan report for proxy.webproxy.local (172.16.2.2)
Host is up (0.00025s latency).
All 1000 scanned ports on proxy.webproxy.local (172.16.2.2) are closed
Nmap done: 1 IP address (1 host up) scanned in 1.63 seconds
```

---

#### MIT FIREWALL

##### BEKANNT OFFENE PORTS

---

##### SSH (PORT 22 TCP)

---

```
web@cli1:~$ sudo nmap -sT -p 22 proxy.webproxy.local
Starting Nmap 6.47 ( http://nmap.org ) at 2016-05-15 17:24 CEST
Nmap scan report for proxy.webproxy.local (172.16.2.2)
Host is up (0.00059s latency).
PORT      STATE SERVICE
22/tcp    filtered ssh
Nmap done: 1 IP address (1 host up) scanned in 0.43 seconds
```

##### WEBSERVER/PROXY(PORT 80 TCP)

---

```
web@cli1:~$ sudo nmap -sT -p 80 proxy.webproxy.local
Starting Nmap 6.47 ( http://nmap.org ) at 2016-05-15 17:25 CEST
Nmap scan report for proxy.webproxy.local (172.16.2.2)
Host is up (0.00047s latency).
```

PORT	STATE	SERVICE
------	-------	---------

**80/tcp filtered http**

Nmap done: 1 IP address (1 host up) scanned in 0.43 seconds

#### DNS(PORT 53 TCP UND UDP)

---

##### TCP

```
web@clil:~$ sudo nmap -sT -p 53 proxy.webproxy.local
```

```
Starting Nmap 6.47 ( http://nmap.org ) at 2016-05-15 17:29 CEST
```

```
Nmap scan report for proxy.webproxy.local (172.16.2.2)
```

```
Host is up (0.00054s latency).
```

PORT	STATE	SERVICE
------	-------	---------

**53/tcp filtered domain**

Nmap done: 1 IP address (1 host up) scanned in 0.43 seconds

##### UDP

```
web@clil:~$ sudo nmap -sU -p 53 proxy.webproxy.local
```

```
Starting Nmap 6.47 ( http://nmap.org ) at 2016-05-15 17:27 CEST
```

```
Nmap scan report for proxy.webproxy.local (172.16.2.2)
```

```
Host is up (0.00053s latency).
```

PORT	STATE	SERVICE
------	-------	---------

**53/udp open|filtered domain**

Nmap done: 1 IP address (1 host up) scanned in 0.43 seconds

#### NMAP AGGRESSIV MODE MIT FIREWALL

---

```
web@clil:~$ sudo nmap -sV -T4 -A proxy.webproxy.local
```

```
Starting Nmap 6.47 ( http://nmap.org ) at 2016-05-15 17:33 CEST
```

```
Nmap scan report for proxy.webproxy.local (172.16.2.2)
```

```
Host is up (0.00036s latency).
```

All 1000 scanned ports on proxy.webproxy.local (172.16.2.2) **are filtered**

Too many fingerprints match this host to give specific OS details

**Network Distance: 2 hops**

TRACEROUTE (using proto 1/icmp)

HOP	RTT	ADDRESS
-----	-----	---------

1	0.21 ms	<b>192.168.3.1</b>
---	---------	--------------------

2	0.29 ms	<b>172.16.2.2</b>
---	---------	-------------------

OS and Service detection performed. Please report any incorrect results at <http://nmap.org/submit/> .

Nmap done: 1 IP address (1 host up) scanned in 24.23 seconds

#### UDP MIT FIREWALL

---

```
web@clil:~$ sudo nmap -sU --open --reason --max-retries 1
```

```
proxy.webproxy.local
```

Starting Nmap 6.47 ( <http://nmap.org> ) at 2016-05-15 18:45 CEST  
Nmap scan report for proxy.webproxy.local (172.16.2.2)  
Host is up, received echo-reply (0.00052s latency).  
All 1000 scanned ports on proxy.webproxy.local (172.16.2.2) **are open|filtered because of 1000 no-responses**  
Nmap done: 1 IP address (1 host up) scanned in 21.37 seconds

#### TCP SYN MIT FIREWALL

---

```
web@cli1:~$ sudo nmap -sS -Pn proxy.webproxy.local
```

Starting Nmap 6.47 ( <http://nmap.org> ) at 2016-05-15 17:36 CEST  
Nmap scan report for proxy.webproxy.local (172.16.2.2)  
**Host is up.**  
All 1000 scanned ports on proxy.webproxy.local (172.16.2.2) **are filtered**  
Nmap done: 1 IP address (1 host up) scanned in 201.33 seconds

#### ACK MIT FIREWALL

---

```
web@cli1:~$ sudo nmap -sA -Pn proxy.webproxy.local
```

Starting Nmap 6.47 ( <http://nmap.org> ) at 2016-05-15 17:45 CEST  
Nmap scan report for proxy.webproxy.local (172.16.2.2)  
**Host is up.**  
All 1000 scanned ports on proxy.webproxy.local (172.16.2.2) **are filtered**  
Nmap done: 1 IP address (1 host up) scanned in 201.33 seconds

#### CONNECT MIT FIREWALL

---

```
web@cli1:~$ sudo nmap -sT -Pn proxy.webproxy.local
```

Starting Nmap 6.47 ( <http://nmap.org> ) at 2016-05-15 17:50 CEST  
Nmap scan report for proxy.webproxy.local (172.16.2.2)  
**Host is up.**  
All 1000 scanned ports on proxy.webproxy.local (172.16.2.2) **are filtered**  
Nmap done: 1 IP address (1 host up) scanned in 201.41 seconds

#### WINDOW MIT FIREWALL

---

```
web@cli1:~$ sudo nmap -sW -Pn proxy.webproxy.local
```

Starting Nmap 6.47 ( <http://nmap.org> ) at 2016-05-15 17:54 CEST  
Nmap scan report for proxy.webproxy.local (172.16.2.2)  
**Host is up.**  
All 1000 scanned ports on proxy.webproxy.local (172.16.2.2) **are filtered**  
Nmap done: 1 IP address (1 host up) scanned in 201.33 seconds

#### MAIMON SCANS MIT FIREWALL

---

```
web@cli1:~$ sudo nmap -sM -Pn proxy.webproxy.local
```

Starting Nmap 6.47 ( <http://nmap.org> ) at 2016-05-15 17:59 CEST

Nmap scan report for proxy.webproxy.local (172.16.2.2)

**Host is up.**

All 1000 scanned ports on proxy.webproxy.local (172.16.2.2) **are filtered**

Nmap done: 1 IP address (1 host up) scanned in 201.33 seconds

---

## TESTUMGEBUNG SICHERHEITSTESTS

---

### OHNE FIREWALL

---

#### NMAP AGRESSIVE MODE OHNE FIREWALL

---

#### UDP OHNE FIREWALL

---

```
web@cli2:~$ sudo nmap -sU --open --reason --max-retries 1
single.webproxy.local
```

Starting Nmap 6.47 ( <http://nmap.org> ) at 2016-05-15 18:51 CEST

Warning: 172.16.2.11 giving up on port because retransmission cap hit (1).

Nmap scan report for single.webproxy.local (172.16.2.11)

Host is up, received echo-reply (0.00042s latency).

All 1000 scanned ports on single.webproxy.local (172.16.2.11) **are open|filtered (908) or closed (92) because of 908 no-responses and 92 port-unreaches**

Nmap done: 1 IP address (1 host up) scanned in 109.10 seconds

---

#### TCP SYN OHNE FIREWALL

---

```
web@cli1:~$ sudo nmap -sS -Pn single.webproxy.local
```

Starting Nmap 6.47 ( <http://nmap.org> ) at 2016-05-15 16:47 CEST

Nmap scan report for single.webproxy.local (172.16.2.11)

Host is up (0.00053s latency).

Not shown: **998 closed ports**

PORT	STATE	SERVICE
------	-------	---------

22/tcp	open	ssh
--------	------	-----

80/tcp	open	http
--------	------	------

Nmap done: 1 IP address (1 host up) scanned in 1.63 seconds

---

#### ACK OHNE FIREWALL

---

```
web@cli1:~$ sudo nmap -sA -Pn single.webproxy.local
```

Starting Nmap 6.47 ( <http://nmap.org> ) at 2016-05-15 17:06 CEST

Nmap scan report for single.webproxy.local (172.16.2.11)

Host is up (0.00075s latency).

All 1000 scanned ports on single.webproxy.local (172.16.2.11) **are unfiltered**

Nmap done: 1 IP address (1 host up) scanned in 1.63 seconds

#### CONNECT OHNE FIREWALL

---

```
web@cli1:~$ sudo nmap -sT -Pn single.webproxy.local
Starting Nmap 6.47 ( http://nmap.org ) at 2016-05-15 17:10 CEST
Nmap scan report for single.webproxy.local (172.16.2.11)
Host is up (0.0032s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
```

Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds

#### WINDOW OHNE FIREWALL

---

```
web@cli1:~$ sudo nmap -sW -Pn single.webproxy.local
Starting Nmap 6.47 ( http://nmap.org ) at 2016-05-15 17:11 CEST
Nmap scan report for single.webproxy.local (172.16.2.11)
Host is up (0.00028s latency).
All 1000 scanned ports on single.webproxy.local (172.16.2.11) are closed
Nmap done: 1 IP address (1 host up) scanned in 1.63 seconds
```

#### MAIMON SCANS OHNE FIREWALL

---

```
web@cli1:~$ sudo nmap -sM -Pn single.webproxy.local
Starting Nmap 6.47 ( http://nmap.org ) at 2016-05-15 17:12 CEST
Nmap scan report for single.webproxy.local (172.16.2.11)
Host is up (0.00032s latency).
All 1000 scanned ports on single.webproxy.local (172.16.2.11) are closed
Nmap done: 1 IP address (1 host up) scanned in 1.64 seconds
```

---

#### MIT FIREWALL

##### NMAP AGRESSIVE MODE MIT FIREWALL

---

```
web@cli1:~$ sudo nmap -sV -T4 -A single.webproxy.local
Starting Nmap 6.47 ( http://nmap.org ) at 2016-05-15 18:05 CEST
Nmap scan report for single.webproxy.local (172.16.2.11)
Host is up (0.00029s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE VERSION
80/tcp    open  http      nginx 1.4.6 (Ubuntu)
| http-auth:
| HTTP/1.1 401 Unauthorized
|_ Basic realm=Restricted
```

|\_http-methods: No Allow or Public header in OPTIONS response (status code 401)

**|\_http-title: 401 Authorization Required**

Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port

Device type: general purpose

Running: Linux 3.X

OS CPE: cpe:/o:linux:linux\_kernel:3

OS details: Linux 3.11 - 3.14

**Service Info: OS: Linux; CPE: cpe:/o:linux:linux\_kernel**

**Network Distance: 2 hops**

TRACEROUTE (using port 80/tcp)

HOP	RTT	ADDRESS
1	0.19 ms	<b>192.168.3.1</b>
2	0.13 ms	<b>172.16.2.11</b>

OS and Service detection performed. Please report any incorrect results at <http://nmap.org/submit/> .

Nmap done: 1 IP address (1 host up) scanned in 21.25 seconds

#### UDP MIT FIREWALL

---

```
web@cli1:~$ sudo nmap -sU --open --reason --max-retries 1
single.webproxy.local
```

Starting Nmap 6.47 ( <http://nmap.org> ) at 2016-05-15 18:38 CEST

Nmap scan report for single.webproxy.local (172.16.2.11)

Host is up, received echo-reply (0.00046s latency).

All 1000 scanned ports on single.webproxy.local (172.16.2.11) **are open|filtered because of 1000 no-responses**

Nmap done: 1 IP address (1 host up) scanned in 21.37 seconds

#### TCP SYN MIT FIREWALL

---

```
web@cli1:~$ sudo nmap -sS -Pn single.webproxy.local
```

Starting Nmap 6.47 ( <http://nmap.org> ) at 2016-05-15 18:10 CEST

Nmap scan report for single.webproxy.local (172.16.2.11)

Host is up (0.00050s latency).

Not shown: **999 filtered ports**

PORT	STATE	SERVICE
------	-------	---------

<b>80/tcp</b>	<b>open</b>	<b>http</b>
---------------	-------------	-------------

Nmap done: 1 IP address (1 host up) scanned in 7.24 seconds

#### ACK MIT FIREWALL

---

```
web@cli1:~$ sudo nmap -sA -Pn single.webproxy.local
Starting Nmap 6.47 ( http://nmap.org ) at 2016-05-15 18:12 CEST
Nmap scan report for single.webproxy.local (172.16.2.11)
Host is up (0.00027s latency).
Not shown: 999 filtered ports
PORT      STATE      SERVICE
80/tcp    unfiltered http
Nmap done: 1 IP address (1 host up) scanned in 7.24 seconds
```

#### CONNECT MIT FIREWALL

---

```
web@cli1:~$ sudo nmap -sT -Pn single.webproxy.local
Starting Nmap 6.47 ( http://nmap.org ) at 2016-05-15 18:13 CEST
Nmap scan report for single.webproxy.local (172.16.2.11)
Host is up (0.00052s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
Nmap done: 1 IP address (1 host up) scanned in 4.35 seconds
```

#### WINDOW MIT FIREWALL

---

```
web@cli1:~$ sudo nmap -sW -Pn single.webproxy.local
Starting Nmap 6.47 ( http://nmap.org ) at 2016-05-15 18:16 CEST
Nmap scan report for single.webproxy.local (172.16.2.11)
Host is up (0.00033s latency).
Not shown: 999 filtered ports
PORT      STATE SERVICE
80/tcp    closed http
Nmap done: 1 IP address (1 host up) scanned in 7.24 seconds
```

#### MAIMON SCANS MIT FIREWALL

---

```
web@cli1:~$ sudo nmap -sM -Pn single.webproxy.local
Starting Nmap 6.47 ( http://nmap.org ) at 2016-05-15 18:17 CEST
Nmap scan report for single.webproxy.local (172.16.2.11)
Host is up.
All 1000 scanned ports on single.webproxy.local (172.16.2.11) filtered
Nmap done: 1 IP address (1 host up) scanned in 201.33 seconds
```

### TEST ANLAGE 7: CLIENT LOAD-BALANCING TESTS

---

#### CLIENTS LOAD-BALANCING TEST SKRIPT

```
#!/bin/bash
```

```
###
```



```

### VARIABLE SETZEN
###
adress=$1
tests_num=$2
re='^[0-9]+$'

echo ""
echo "Testen verbindungen..."
if [ -z $adress ]
then
    echo "Keine Adresse eingegeben..."
    echo "Bitte benutze es: ./test_verbindung <ADRESSE> <NUM_TEST>"
elif [ -z $tests_num ]
then
    echo "Keine zahl für tests eingegeben..."
    echo "Bitte benutze es: ./test_verbindung <ADRESSE> <NUM_TEST>"
else

    if ! [[ $tests_num =~ $re ]]
    then
        echo "Eingabe für test soll ein number sein: $tests_num" >&2
    else

        echo "Machen $tests_num tests aufruff zur $adress..."
        echo ""
        for i in `seq 1 $tests_num`;
        do
            #echo $i
            curl $adress
        done
        echo ""
        echo "Fertig..."
        echo ""

    fi

fi

```

---

#### ROUND-ROBING LOAD-BALANCING TESTS

```

web@cli1 ~ $ ./test_verbindung web:<USER_PASSWD>@round.webproxy.local 15
Testen verbindungen...
Machen 15 tests aufruff zur web:<USER_PASSWD>round.webporxy.local...

apache1

```

apache2  
apache3  
apache1  
apache2  
apache3  
apache1  
apache2  
apache3  
apache1  
apache2  
apache3  
apache1  
apache2  
apache3

Fertig ...

---

#### WEIGHTED LOAD-BALANCING TESTS

```
web@cli1:~$ ./test_verbindung web:<USER_PASSWD>@weighted.webproxy.local 15
Testen verbindungen...
15 testen aufruff zur web:<USER_PASSWD>@weighted.webproxy.local...
```

apache 1  
apache 2  
apache 1  
apache 3  
apache 1  
apache 1  
apache 2  
apache 1  
apache 3  
apache 1  
apache 1  
apache 2  
apache 1  
apache 3  
apache 1

Fertig...

---

#### PERSISTENT LOAD-BALANCING TESTS

```
web@cli1:~$ ./test_verbindung web:<USER_PASSWD>@persistent.webproxy.local  
15
```

Testen verbindungen...

Machen 15 testen aufruff zur web:<USER\_PASSWD>@persistent.webproxy.local...

```
apache 1  
apache 1  
apache 1  
apache 1  
apache 1  
apache 1  
apache 1  
apache 1  
apache 1  
apache 1  
apache 1  
apache 1  
apache 1  
apache 1  
apache 1
```

Fertig...

---

#### LEAST-CONNECTED LOAD-BALANCING TESTS

```
web@cli1:~$ ./test_verbindung web:<USER_PASSWD>@least.webproxy.local 15
```

Testen verbindungen...

Machen 15 testen aufruff zur web:<USER\_PASSWD>@least.webproxy.local...

```
apache 1  
apache 2  
apache 3  
apache 1  
apache 2  
apache 3  
apache 1  
apache 2  
apache 3  
apache 1  
apache 2
```

apache 3  
apache 1  
apache 2  
apache 3

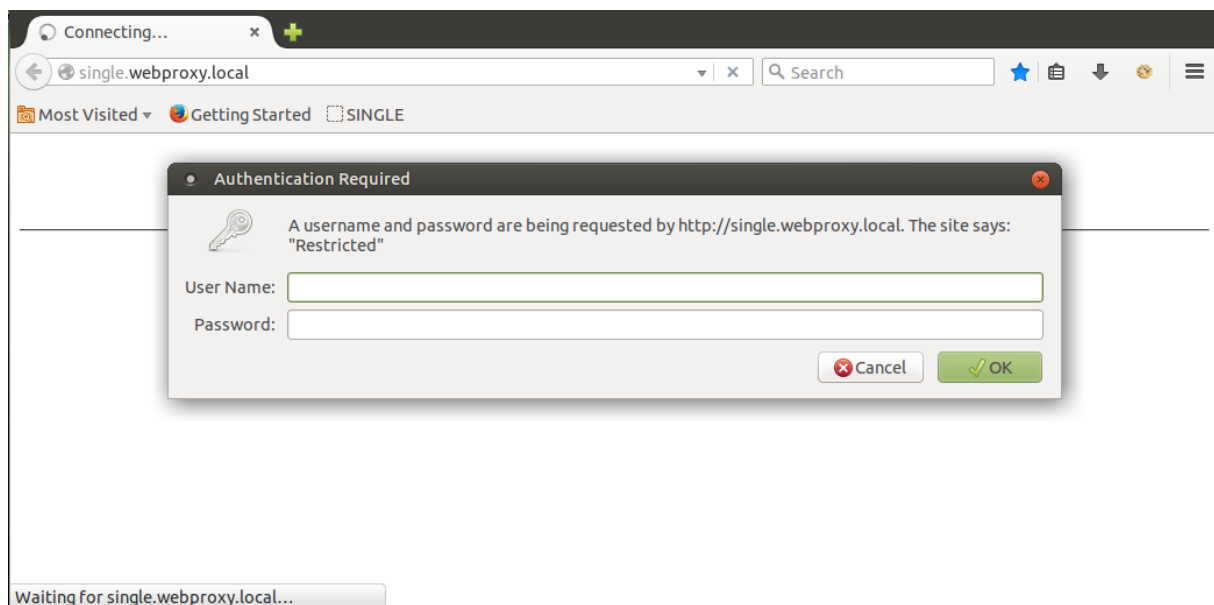
Fertig...

## KUNDENDOKUMENTATION ANLAGE

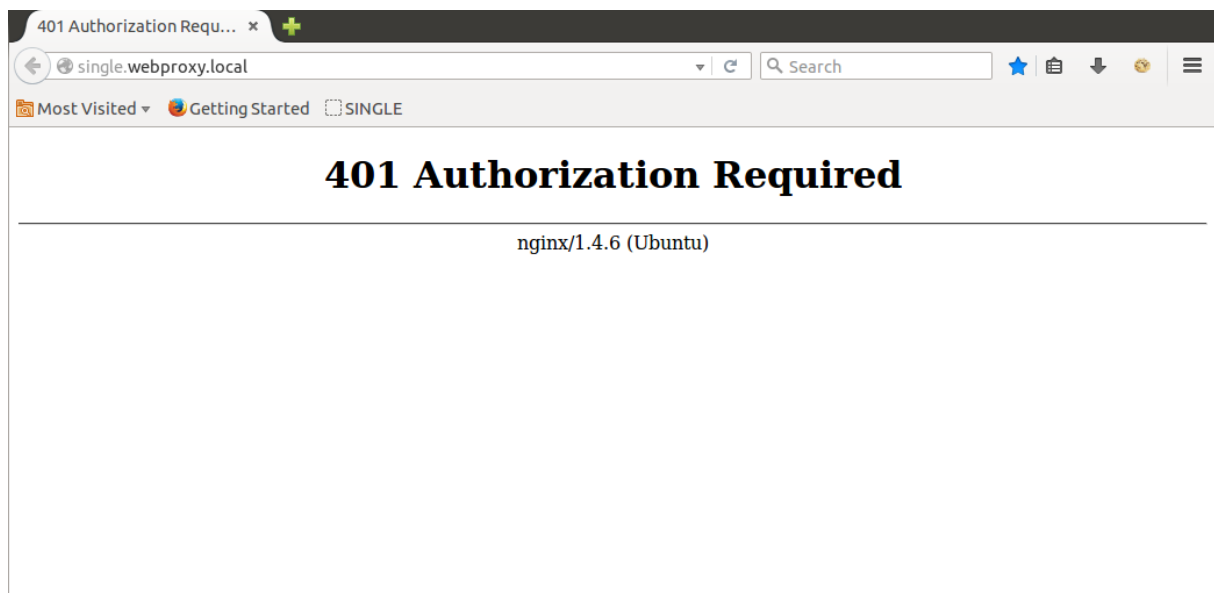
### TESTUMGEBUNG

#### EINLOGGEN

Hier sollten Sie Ihren Benutzernamen und Ihr Passwort eingeben.



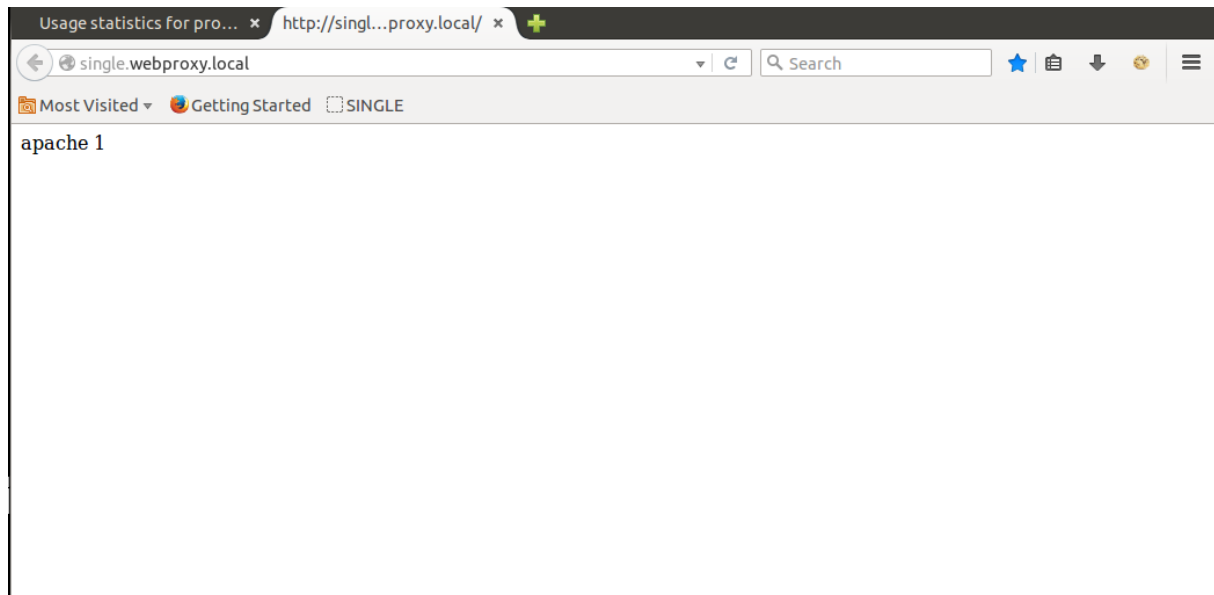
#### NICHT AUTHORIZIERT



---

## WEB INHALT NACH AUTHORISIERUNG

nach, dass Sie könnte die Anmeldung sollten Sie den Inhalt lesen können.



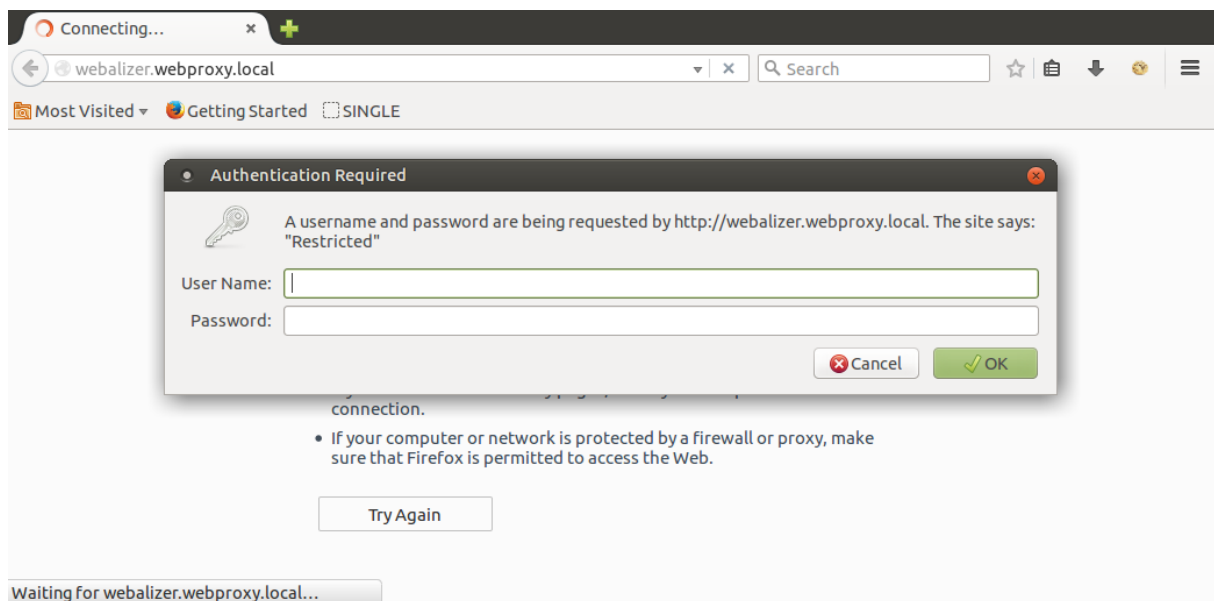
---

## WEBALIZER

---

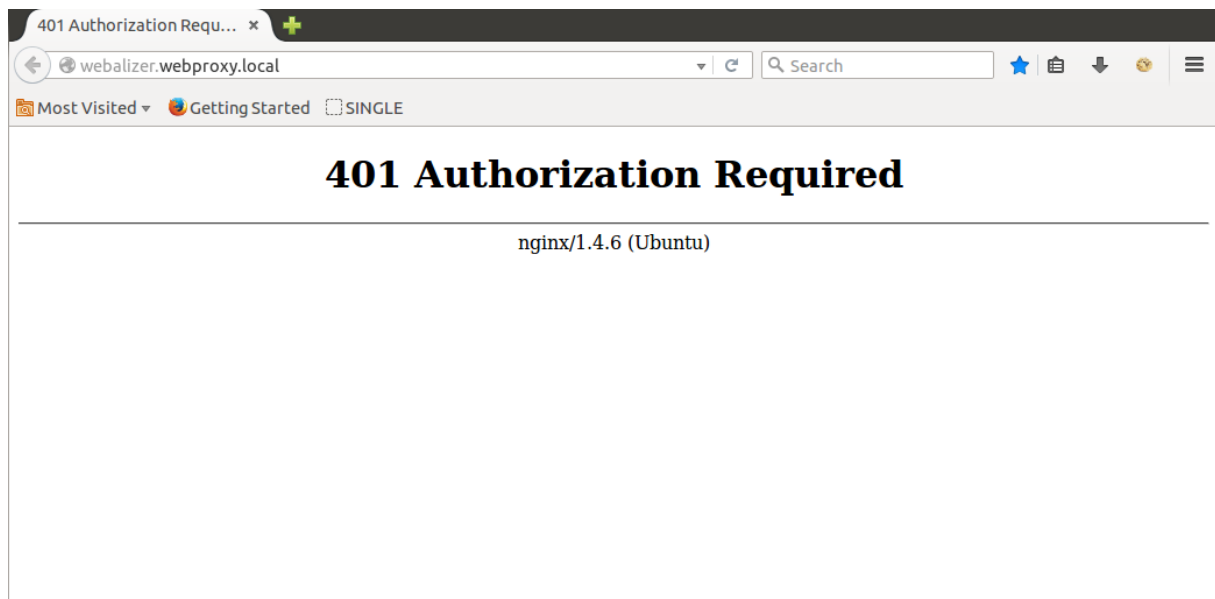
### EINLOGGEN

Hier sollten Sie Ihren Benutzernamen und Ihr Passwort eingeben.

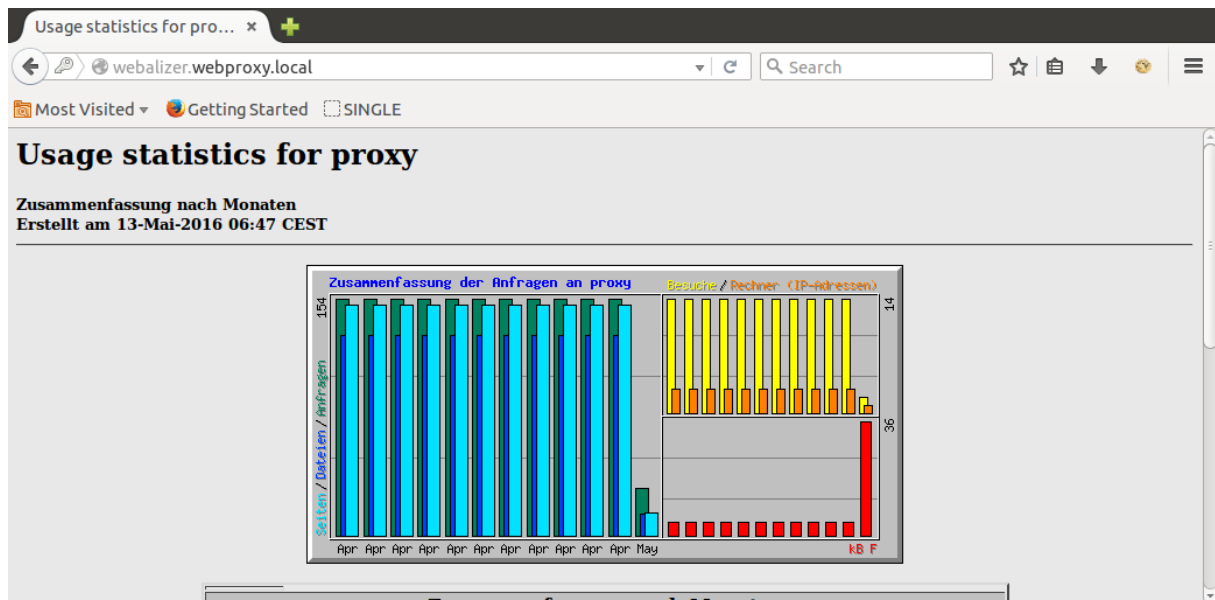


---

### NICHT AUTHORIZIERT



## WEBALIZER ANWENDUNG NACH AUTHORIZIERUNG



Usage statistics for proxy

Auslastungsstatistik für den Zeitraum: May 2016  
Erstellt am 13-Mai-2016 06:47 CEST

[\[Tages-Statistik\]](#) [\[Stunden-Statistik\]](#) [\[URLs\]](#) [\[Eingang\]](#) [\[Ausgang\]](#) [\[Rechner \(IP-Adressen\)\]](#) [\[Verweise\]](#) [\[Suche\]](#) [\[Benutzer\]](#) [\[Anwenderprogramme\]](#) [\[Länder\]](#)

Monats-Statistik für May 2016	
Summe Anfragen	31
Summe Dateien	14
Summe Seiten	15
Summe Besuche	2
Total kB Files	36
Summe unterschiedlicher Rechner (IP-Adressen)	1
Summe unterschiedlicher URLs	6
Summe unterschiedlicher Verweise	3
Summe unterschiedlicher Benutzer	1
Summe unterschiedlicher Anwenderprogramme	2

## ADMIN-DOKUMENTATION ANLAGE

Nach einer erfolgreichen Verbindung zum Proxy-Server wird ein Scripts ausgeführt. Wird das Menüs angezeigt.

PROXY SERVER MENU

- 1 Users
- 2 Servers
- 3 Containers

Option:

### 1 BENUTZER

Bei der Benutzerfunktionalität wird das Untermenü angezeigt.

USERS MENU

- 1 Show Users
- 2 Add User
- 3 Update User
- 4 Delete User

Option:

---

#### 1.1 BENUTZER EINSETZEN (OPTIONEN 2)

Mit dieser Option wird ein neuer Benutzer zum System hinzugefügt. Der Benutzername und das Passwort sollte hier angegeben werden.

USERS MENU

- 1 Show Users
- 2 Add User
- 3 Update User
- 4 Delete User

Option: **2**

Adding User...

ADD NEW USER

Please enter new user name: test\_user

New user name: **test\_user**

Is that correct (Y/N): y

RUN COMMAND...

New password:

Re-type new password:

Adding password for user test\_user

---

#### 1.2 BENUTZERN AUSDRÜCKEN (OPTIONEN 1)

Mit dieser Option können alle Benutzer auf dem System angezeigt werden.

USERS MENU

- 1            Show Users
- 2            Add User
- 3            Update User
- 4            Delete User

Option: **1**

Show Users...

web

**test\_user**

---

### 1.3 EXISTIERENDEN BENUTZER AKTUALISIEREN (OPTIONEN 3)

Mit dieser Option, dass der Benutzer bereits auf dem System vorhanden ist, kann das Passwort aktualisiert werden.

USERS MENU

- 1            Show Users
- 2            Add User
- 3            Update User
- 4            Delete User

Option: **3**

Updating User...

UPDATE USER

Please user name: **test\_user**

User name: test\_user

Is that correct(Y/N): y

UPDATE USER: test\_user

New password:

Re-type new password:

Updating password for user test\_user

---

### 1.4 BENUTZER LOSCHEN (OPTIONEN 4)

Bei dieser Option wird der Benutzer, der in dem System vorhanden ist, gelöscht.

USERS MENU

- 1            Show Users
- 2            Add User
- 3            Update User
- 4            Delete User

Option: **4**

Deleting User...



DELETE USER

Please user name: **test\_user**

User name: test\_user

Is that correct(Y/N): y

Deleting password for user test\_user

---

## 1.5 BENUTZERN AUSDRÜCKEN (OPTIONEN 1)

Um zu bestätigen, dass der gelöschte Benutzer nicht mehr auf das System ist, wird die Show-Funktion erneut aufgerufen.

USERS MENU

- 1            Show Users
- 2            Add User
- 3            Update User
- 4            Delete User

Option: **1**

Show Users...

web

## 2 SERVERS

Das Servermenü wird angezeigt.

SERVER MENU

- 1            Show Servers
- 2            Add Server
- 3            Delete Server

Option:

---

### 2.1 SERVER EINSETZEN (OPTIONEN 2)

Bei dieser Funktion werden Sie zum Untermenü gebracht die für die Wahl steht, welche Art von Server zu dem System hinzugefügt werden soll.

SELECT SERVER TYPE MENU

- 1            Add as Single Server
- 2            Add Server to Load-Balancing

Option:

---

#### 2.1.1 SERVER NICHT INS LOAD-BALANCING EINSETZEN

Hier wird das Untermenü angezeigt., und nachgefragt nach dem Servernamen und IP-Adresse.

SELECT SERVER TYPE MENU

- 1            Add as Single Server
- 2            Add Server to Load-Balancing

Option: **1**

Adding as Single Server...

Single Server name to be added: **test\_server\_1**

New server name: test\_server\_1.webproxy.local

Is that correct(Y/N): y

Please enter the IP adress: **10.0.0.22**

New server IP: 10.0.0.22

Is that correct(Y/N): y

---

### 2.1.2 SERVER INS LOAD-BALANCING EINSETZEN

Hier wird das Untermenü angezeigt., und nachgefragt nach dem Servernamen und IP-Adresse..

Hier wird auch angezeigt welche art von Load-Balacing zugefügt wird.

ADD TO LOAD-BALANCING MENU

- 1            Add to Round-Robin
- 2            Add to Weighted
- 3            Add to Persistent
- 4            Add to Least Connected

Option: **2**

Adding to Weighted...

Server to be added to: **weighted.webproxy.local**

Is that correct(Y/N): y

Please enter the IP adress: **10.0.0.22**

New server IP: 10.0.0.22

Is that correct(Y/N): y

ADD SERVER TO BALANCING: **weighted**

Want to add weight-value to the server?(Y/N): y

Enter weight-value: **5**

New weight-value: 5

Is that correct(Y/N): y

---

### 2.2 SERVER AUSDRÜCKEN (OPTIONEN 1)

Nach der Wahl der Funktionalität werden alle Server angezeigt.

SERVER MENU

- 1            Show Servers
- 2            Add Server
- 3            Delete Server

Option: **1**

Show Servers...

SERVER\_NAME: **test\_server\_1**

TYPE: **single\_server**

IP: **10.0.0.22**

PROXY: **172.16.2.100**

ADAPTER: **eth1:added1**

SERVER\_NAME: **weighted\_1**

TYPE: load-balancing

IP: **10.0.0.22**

PROXY: 172.16.2.13

ADDRESS: weighted.webproxy.local

---

## 2.3 SERVER LÖSCHEN (OPTIONEN 3)

Das Untermenü wird angezeigt.

SELECT SERVER TYPE MENU TO DELETE

1 Delete Single Server

2 Delete Server from Load-Balancing

Option:

---

### 2.3.1 SERVER NICHT INS LOAD-BALANCING LÖSCHEN

Hier wird der Name des Servers gefragt, der gelöscht werden soll.

SELECT SERVER TYPE MENU TO DELETE

1 Delete Single Server

2 Delete Server from Load-Balancing

Option: **1**

Delete Single Server...

Single Server name to be deleted: **test\_server\_1**

Server address: test\_server\_1.webproxy.local

Is that correct(Y/N): y

Hier wird wieder die SHOW Funktionalität angezeigt um zu lesen das der Server gelöscht ist.

SERVER MENU

1 Show Servers

2 Add Server

3 Delete Server

Option: **1**

Show Servers...

SERVER\_NAME: **weighted\_1**

TYPE: load-balancing

IP: **10.0.0.22**

PROXY: 172.16.2.13

ADDRESS: weighted.webproxy.local

---

### 2.3.2 SERVER INS LOAD-BALANCING LÖSCHEN

Hier wird nachgefragt welche Server gelöscht werden soll.

SELECT SERVER TYPE MENU TO DELETE

1 Delete Single Server

2 Delete Server from Load-Balancing

Option: **2**

Delete Server from Load-Balancing...

Load-Balancing Server name to be deleted: **weighted\_1**

Server name: weighted\_1

Is that correct (Y/N): y

Hier wird wieder die SHOW Funktionalität angezeigt um zu lesen das der Server gelöscht ist.

SERVER MENU

1 Show Servers

2 Add Server

3 Delete Server

Option: **1**

Show Servers...

No added servers found...

## 3 CONTAINERS

Hier wird das Containermenü angezeigt.

CONTAINERS MENU

1 Show Containers

2 Add Container

3 Delete Container

Option:

---

### 3.1 CONTAINERS EINSETZEN (OPTIONEN 2)

Bei dieser Funktion werden Sie zum Containermenü gebracht die für die Wahl steht, welche Art von Container zu dem System hinzugefügt werden soll.

SELECT CONTAINER TYPE MENU

1 Add as Single Server

2 Add Container to Load-Balancing

Option:

---

### 3.1.1 CONTAINER NICHT INS LOAD-BALANCING EINSETZEN

Hier wird das Untermenü angezeigt., und nachgefragt nach dem Containernamen,Containerserver von IP-Adresse, und Container Portnr..

```
SELECT CONTAINER TYPE MENU
```

- 1            Add as Single Server
- 2            Add Container to Load-Balancing

Option: **1**

Adding as Single Server...

Single Container name to be added: **container\_test**

New server name: container\_test.webproxy.local

Is that correct(Y/N): y

Please enter the Container-Server IP adress: **10.0.0.44**

Container-Server IP: 10.0.0.44

Is that correct(Y/N): y

Please enter the Container port number: **4001**

Container port: 4001

Is that correct(Y/N): y

---

### 3.1.2 CONTAINERS INS LOAD-BALANCING EINSETZEN

Hier wird das Untermenü angezeigt., und nachgefragt nach dem Containernamen,Containerserver von IP-Adresse, und Container Portnr..

Hier wird auch angezeigt welche art von Load-Balacing zugefügt wird.

```
ADD CONTAINER TO LOAD-BALANCING MENU
```

- 1            Add to Round-Robin
- 2            Add to Weighted
- 3            Add to Persistent
- 4            Add to Least Connected

Option: **2**

Adding to Weighted...

Server to be added to: c-weighted.webproxy.local

Is that correct(Y/N): y

Please enter the Container-Server IP adress: **10.0.0.44**

Container-Server server IP: 10.0.0.44

Is that correct(Y/N): y

Please enter the Container port number: **4001**

Container port: 4001

Is that correct(Y/N): y

```
ADD SERVER TO BALANCING: c-weighted
Want to add weight-value to the server?(Y/N): y
Enter weight-value: 5
New weight-value: 5
Is that correct(Y/N): y
```

---

### 3.2 CONTAINERS AUSDRÜCKEN (OPTIONEN 1)

Nach der Wahl der Funktionalität werden alle Container angezeigt.

CONTAINERS MENU

- 1            Show Containers
- 2            Add Container
- 3            Delete Container

Option: **1**

Show Containers...

CONTAINER\_NAME: **container\_test**

TYPE: single\_container

IP: **172.16.2.101**

PORT: **4001**

PROXY: **10.0.0.44**

ADAPTER: **eth1:added2**

CONTAINER\_NAME: **c-weighted\_1**

TYPE: load-balancing

IP: **10.0.0.44**

PORT: **4001**

PROXY: 172.16.2.23

ADDRESS: c-weighted.webproxy.local

---

### 3.3 CONTAINERS LÖSCHEN (OPTIONEN 3)

Das Untermenü wird angezeigt.

SELECT CONTAINER TYPE MENU

- 1            Delete Single Server Container
- 2            Delete Container from Load-Balancing

Option:

---

#### 3.3.1 CONTAINERS NICHT INS LOAD-BALANCING LÖSCHEN

Hier wird der Name des Container gefragt, der gelöscht werden soll.

SELECT CONTAINER TYPE MENU

- 1            Delete Single Server Container
- 2            Delete Container from Load-Balancing

Option: **1**

Delete Single Container...

Single Server container name to be deleted: **container\_test**

Container Server address: container\_test.webproxy.local

Is that correct(Y/N): y

Hier wird wieder die SHOW Funktionalität angezeigt um zu lesen das der Container gelöscht ist.

CONTAINERS MENU

- 1 Show Containers
- 2 Add Container
- 3 Delete Container

Option: **1**

Show Containers...

CONTAINER\_NAME: c-weighted\_1

TYPE: load-balancing

IP: 10.0.0.44

PORT: 4001

PROXY: 172.16.2.23

ADDRESS: c-weighted.webproxy.local

---

### 3.3.2 CONTAINERS INS LOAD-BALANCING LÖSCHEN

Hier wird nachgefragt welche Container gelöscht werden soll.

SELECT CONTAINER TYPE MENU

- 1 Delete Single Server Container
- 2 Delete Container from Load-Balancing

Option: **2**

Delete Container from Load-Balancing...

Load-Balancing Container name to be deleted: **c-weighted\_1**

Container name: c-weighted\_1

Is that correct(Y/N): y

Hier wird wieder die SHOW Funktionalität angezeigt um zu lesen das der Server gelöscht ist.

CONTAINERS MENU

- 1 Show Containers
- 2 Add Container
- 3 Delete Container

Option: **1**

Show Containers...

No added containers found...



## LITERATURVERZEICHNIS

Linux-Command.org. (2016). Von [http://linuxcommand.org/man\\_pages](http://linuxcommand.org/man_pages) abgerufen

NginX. (2016). *Load-Balancing*. Von [http://nginx.org/en/docs/http/load\\_balancing.html](http://nginx.org/en/docs/http/load_balancing.html) abgerufen

Wikipedia. (2016). Von <https://de.wikipedia.org> abgerufen

## Floxy System Flussdiagramm

