

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

Einleitung

Install und Konfig

Tests

1

2

3

4

5

6

Planung

Implementierung

Abschluss

Einleitung

Install und Konfig

Tests

1

2

3

4

5

6

Planung

Implementierung

Abschluss

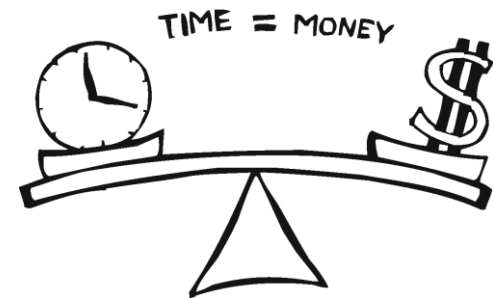


- 1.1- Bruno Reboucas
- 1.2- Projektumfeld
 - Firma name: Sage Software GmbH
 - Firma Aufgaben: Entwicklung von kaufmännischer Software
 - Buchhaltung
 - Lohnsteuer
 - Steuerrechnung
 - u.a.

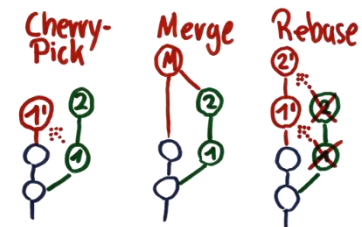


1.3- Projektbeschreibung

- Zeitaufwand für spezifische Tests der Entwickle zu hoch



- Testspezifikationen zu teilen kompliziert





1.4 Projektbegründung

- Einrichtung einer für alle Entwickler erreichbare Testumgebung
- Testumgebung muss plattform- und betriebssystemunabhängig sein



1.5- Projektziel

- Keine Änderung am bestehenden Netzwerk
- Skalierbar
- Beliebig konfigurierbar
- Niedrige Kosten
- Verbindung zu allen Testumgebungen durch einen transparenten Webproxy ermöglichen



1.6- Personlicheschnittstelle

- Verantwortlich für Internes Netzwerk: Herr Petsching

Einleitung

Install und Konfig

Tests

1

2

3

4

5

6

Planung

Implementierung

Abschluss

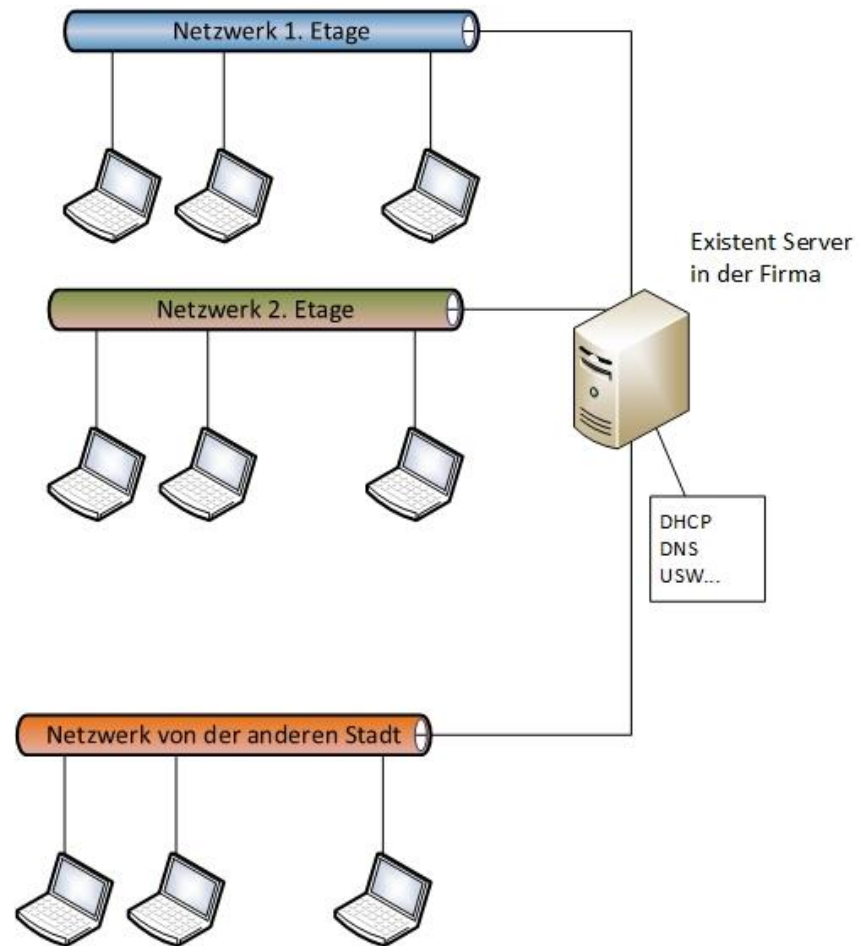


2.1- Anforderung

- Zentralisiert Zugriff auf der Testumgebungen
 - Identifizierung/Authentifizierung
- Systemübergreifend
- Skalierbar
- Load-Balancing (Später dazu gekommen)
- Absicherung
 - Firewall/DMZ

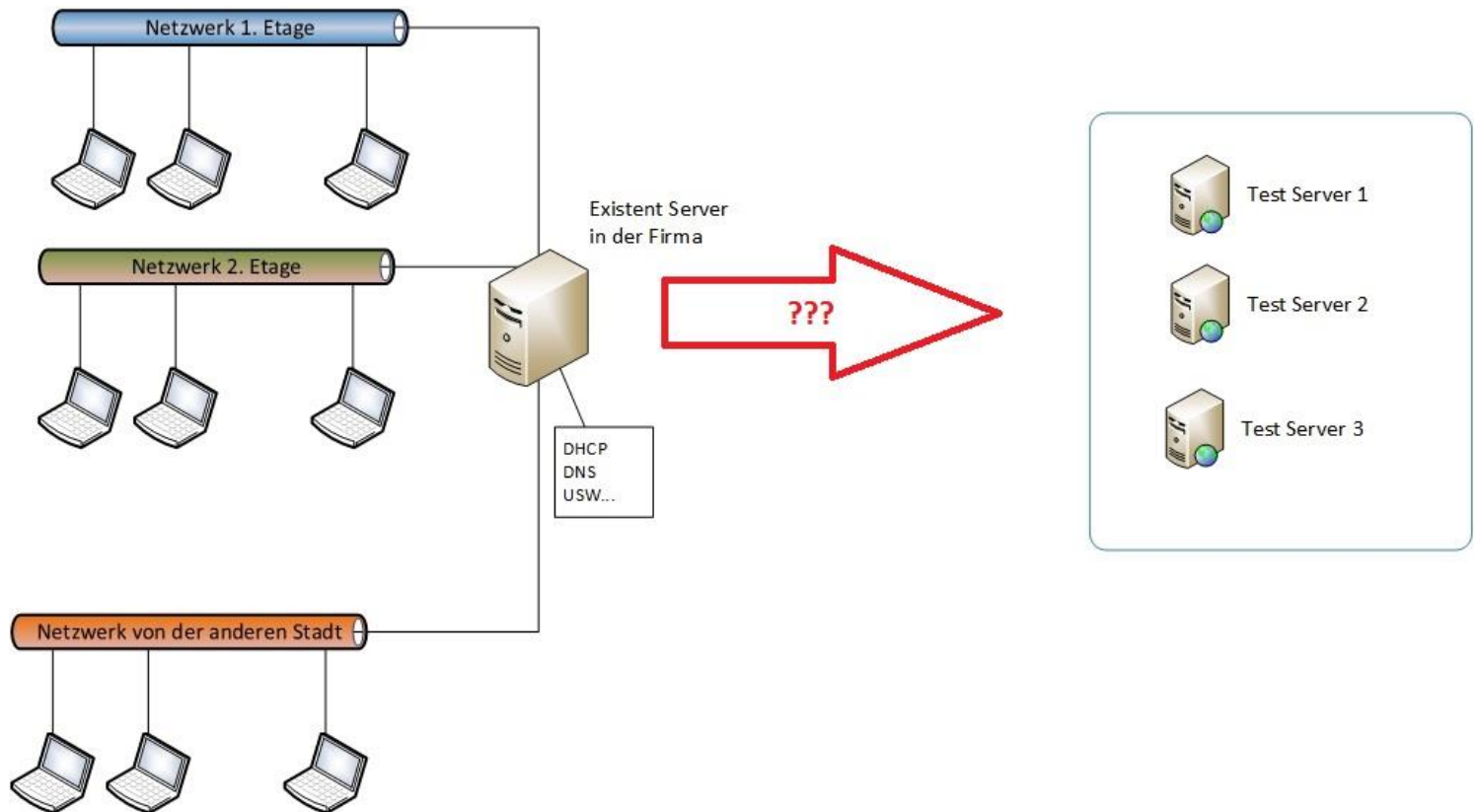


2.2- Ist-Zustand



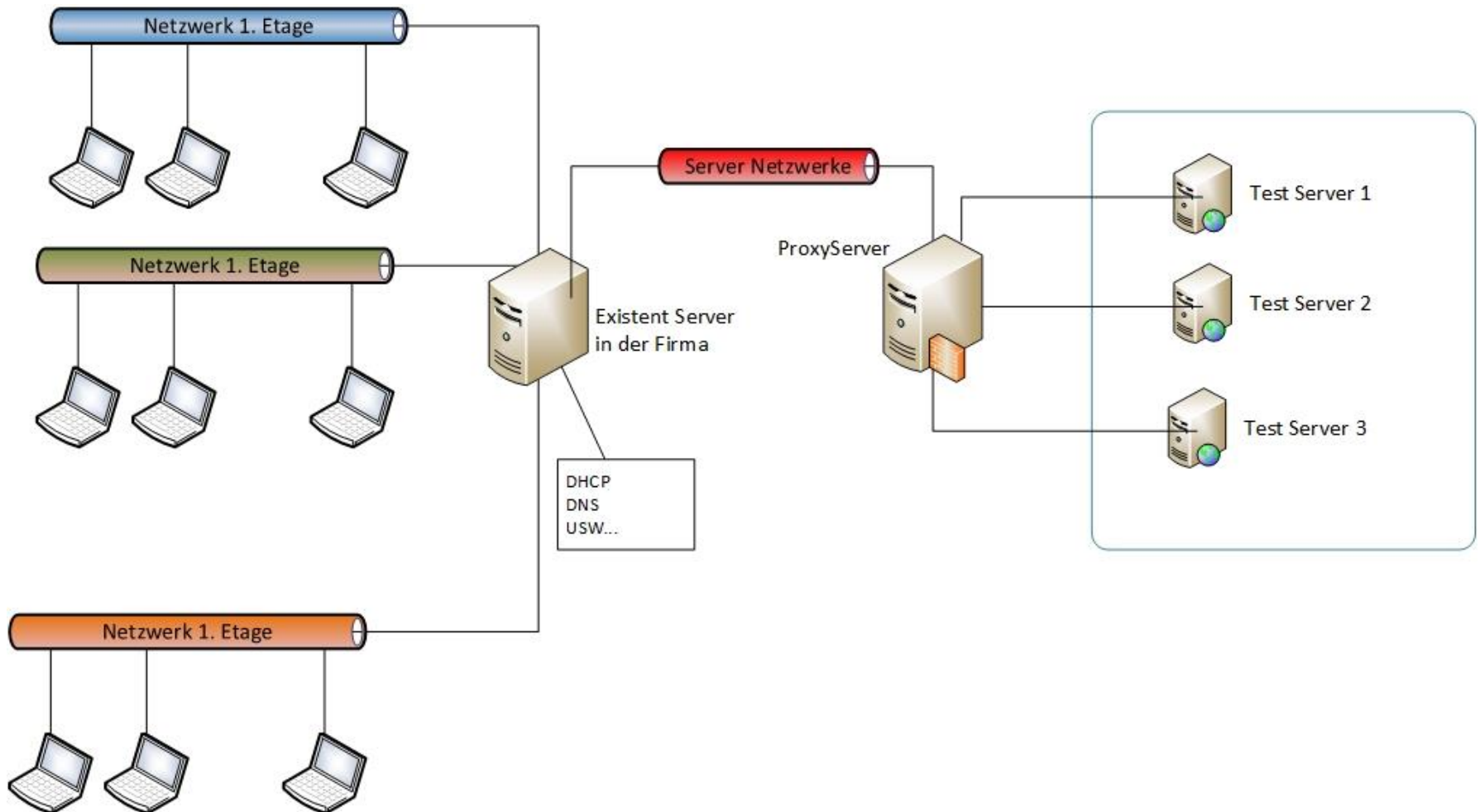


2.3- Problembeschreibung



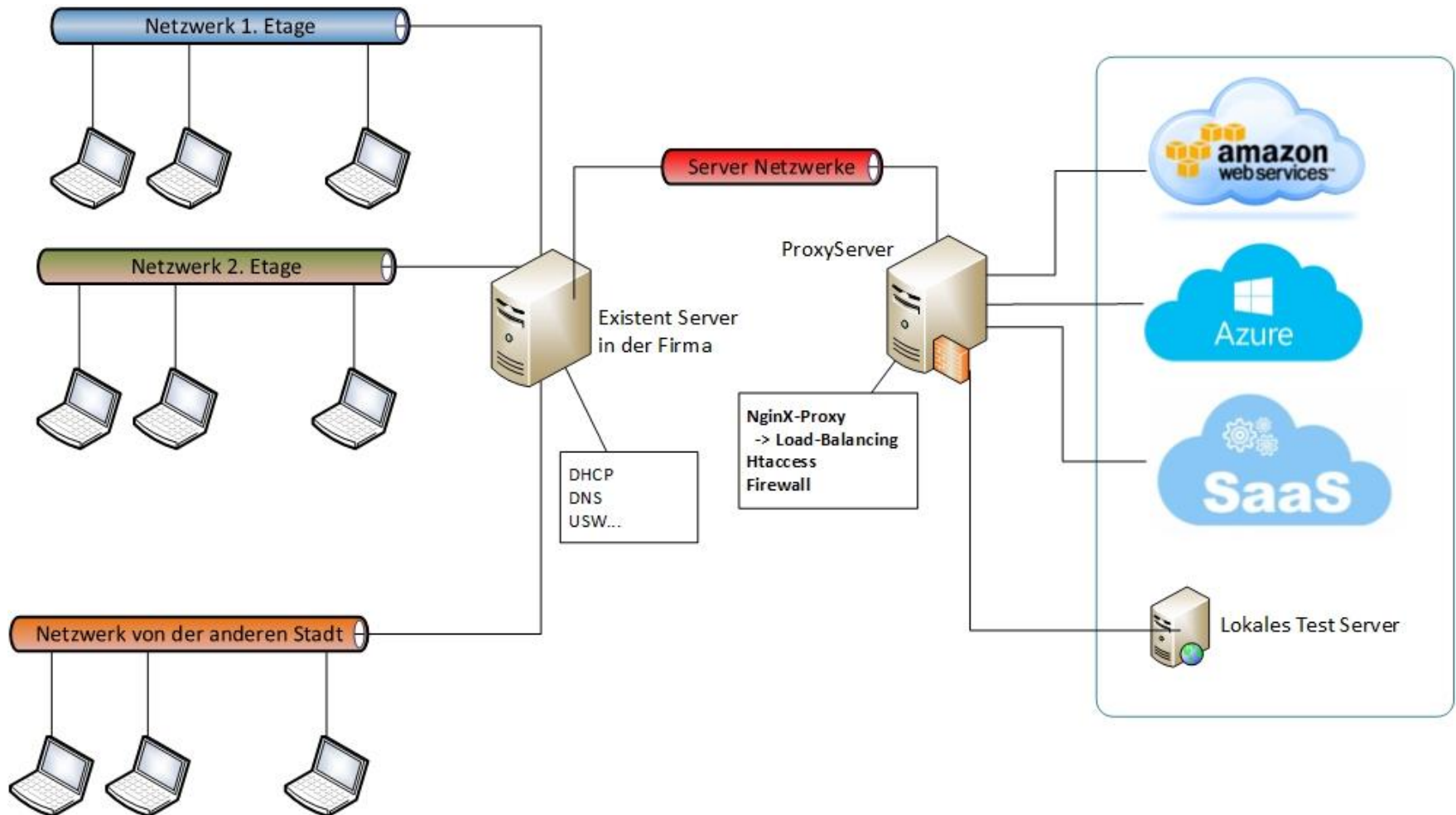


2.4- Lösungsansatz



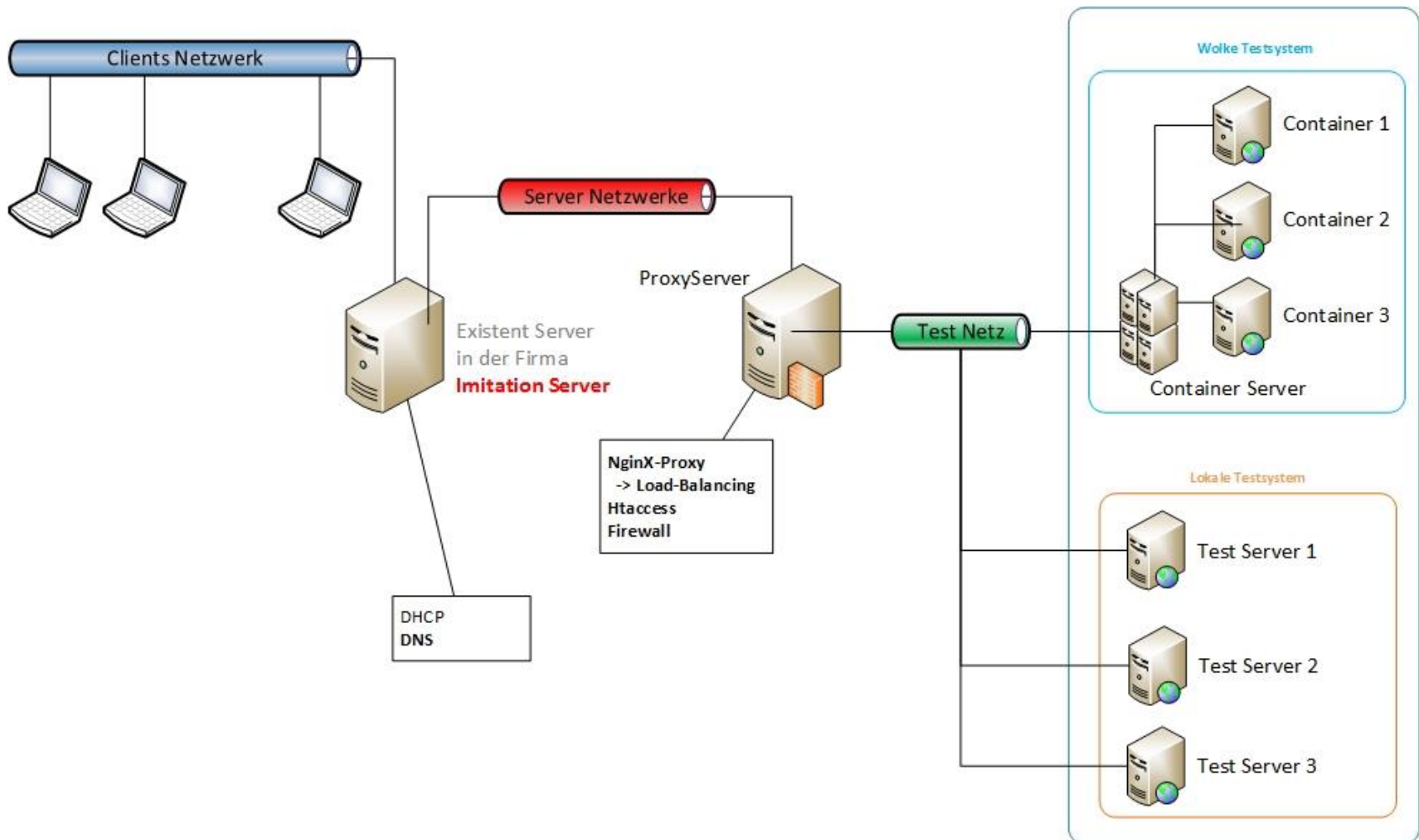


2.5- Gewünscht

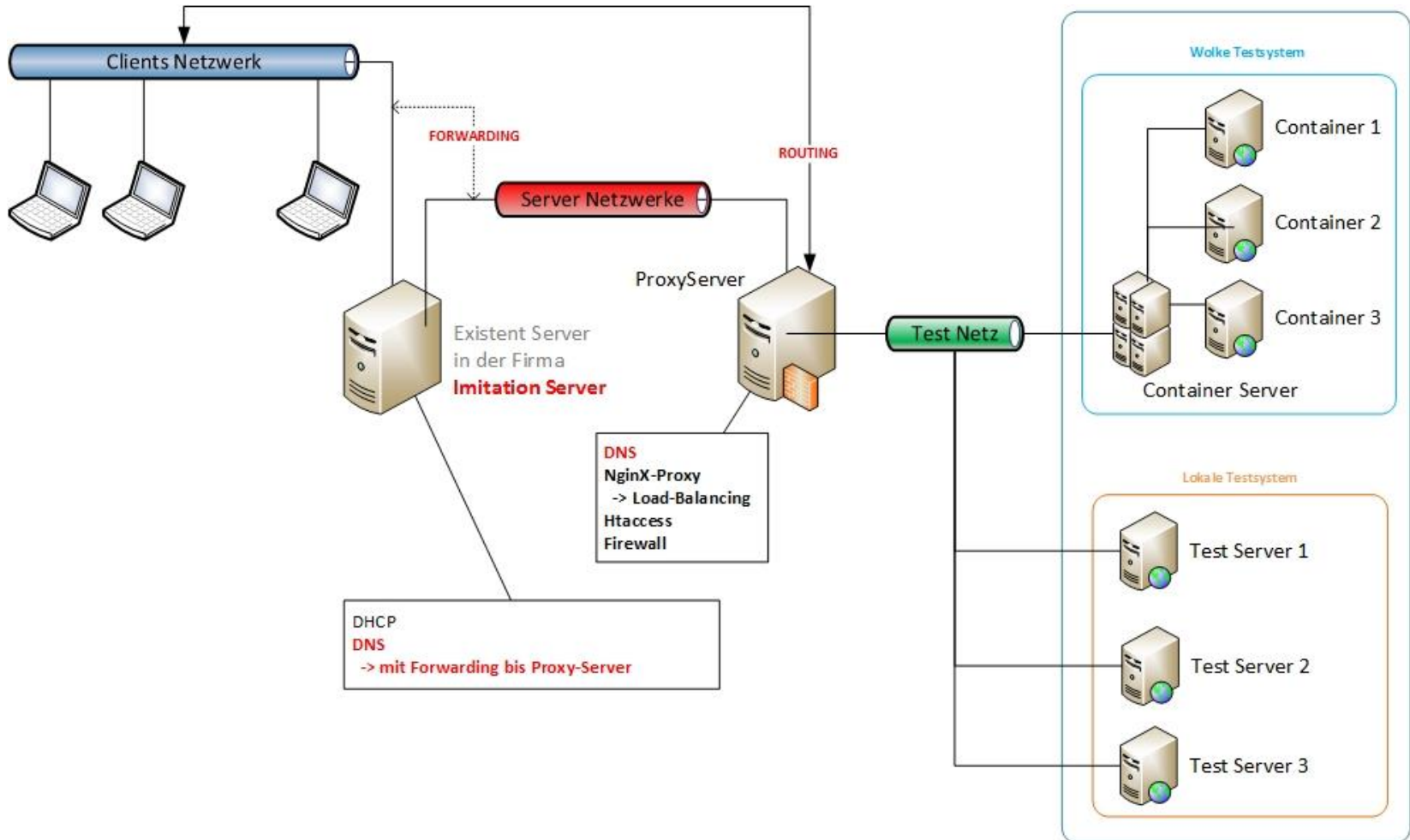




2.6- Soll-Zustand: Dieses Projekt

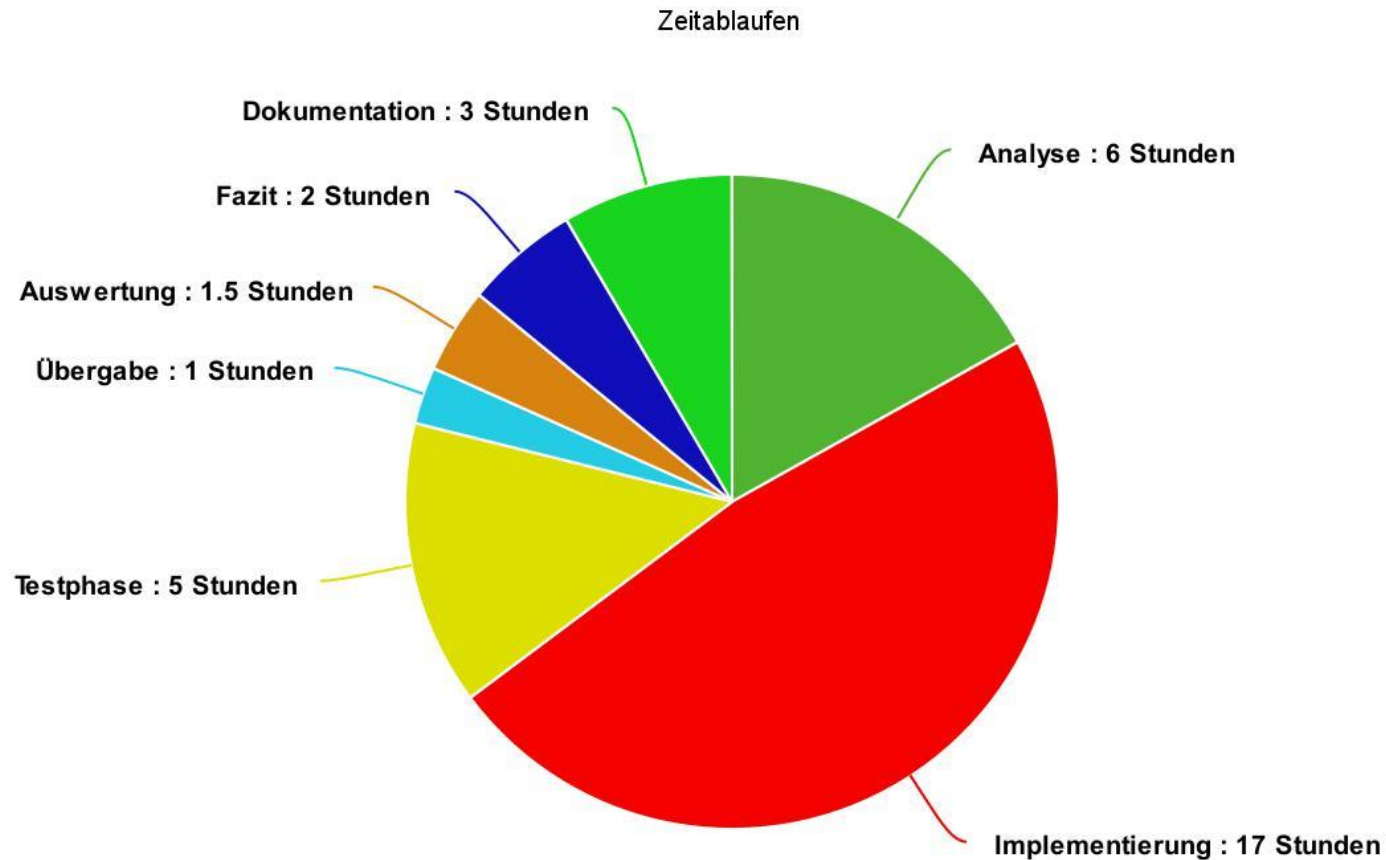


2.7- Soll-Zustand: Dieses Projekt Netzwerk Problemen





2.8- Projektphase: Zeitablaufen





2.9- Projektkosten

- Kostengünstig
- Lizenzfrei
- Nur Lohnkosten:
 - $60\text{€} \times 35\text{h} = 2100\text{€}$

Einleitung

Install und Konfig

Tests

1

2

3

4

5

6

Planung

Implementierung

Abschluss



3.1- Installation und Konfiguration: Test Clients

- Verschiedenen Systemen (Windows und Linux)

3.2- Installation und Konfiguration: Web Servers

- Regular Webserver
 - Apache oder NginX
- Container Server
 - Webserver Container
 - NginX



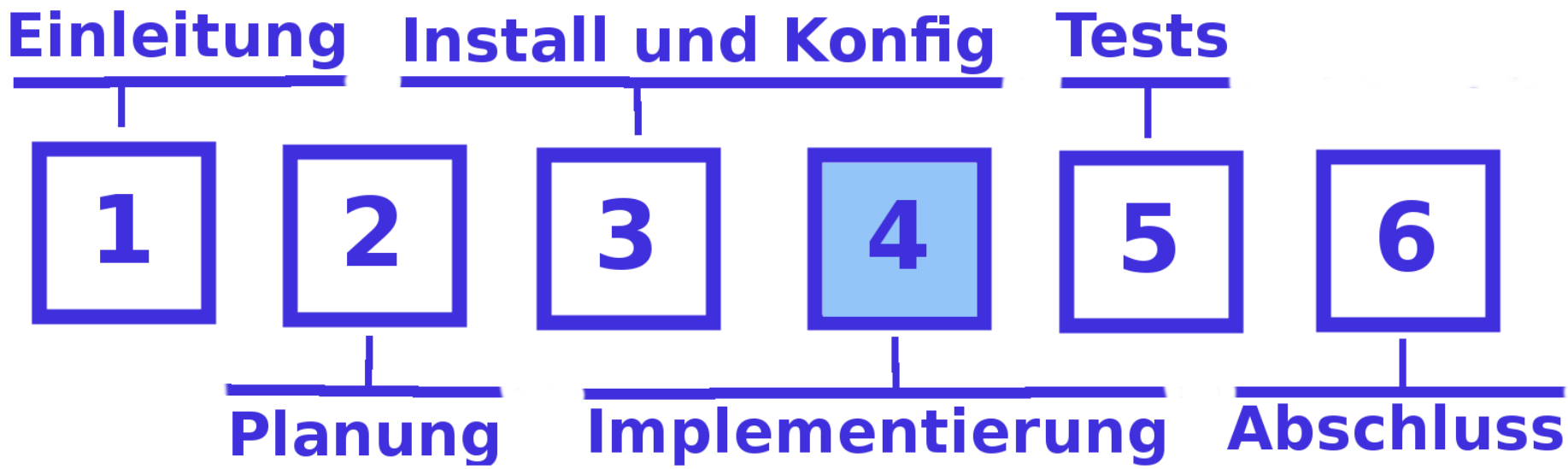
3.3- Installation und Konfiguration: Imitation Server

- DHCP
- DNS
 - DNS auth
 - Port-Forwarding nach ProxyServer DNS
- Port weiterleitung
 - Nach intern Netzwerk



3.4- Installation und Konfiguration: Proxy Server

- DNS
 - Authentifizierung
- NginX
 - Load-Balancing
 - Webalizer
 - Log-Datei Auswerten
- Htaccess
 - Benutzer auth
- Firewall/DMZ
 - Zugang nur für erlaubtes Netzwerk
 - Auto-reload wenn ein neuer Server online geht
- Alles on-the-fly aktualisieren





5.1- Benutzer

- Zugang durch .htaccess
 - Einsetzen
 - Aktualisieren
 - Löschen



- 5.2- Single Server

- Von einem Server

- IP
 - Domain Name: beispiel-server-1
 - beispiel-server-1.webproxy.local
 - Firewall update mit neuer virtuellen Netzwerkschnittstelle

- Von einem Container

- Container Server IP
 - Container Port
 - Firewall Update



5.3 - Load-Balancing Server

– Von Server

- IP
- Welche Art von Load-Balancing
- Firewall update

– Von Container

- Container Server IP
- Container port
- Welche Art von Load-Balancing
- Firewall update

– ZUKUNFTIGE ERWEITERUNG: filtern nach Port

Einleitung

Install und Konfig

Tests

1

2

3

4

5

6

Planung

Implementierung

Abschluss



5.1- Verbindung Tests: Single Server

```
web@cli4:~$ traceroute single.webproxy.local
traceroute to single.webproxy.local (172.16.2.11), 30 hops max, 60 byte packets
 1 192.168.3.1 (192.168.3.1) 0.216 ms 0.171 ms 0.133 ms
 2 172.16.2.11 (172.16.2.11) 1.141 ms 1.028 ms 0.982 ms
web@cli4:~$ 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>
web@cli4:~$ curl web:1234@single.webproxy.local
apache 1
web@cli4:~$ traceroute c-single.webproxy.local
traceroute to c-single.webproxy.local (172.16.2.21), 30 hops max, 60 byte packets
 1 192.168.3.1 (192.168.3.1) 0.175 ms 0.169 ms 0.087 ms
 2 172.16.2.21 (172.16.2.21) 0.669 ms 0.627 ms 0.595 ms
web@cli4:~$ curl c-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>
web@cli4:~$ curl web:1234@c-single.webproxy.local
nginx_1 DOCKER
web@cli4:~$
```

Imitation Server
Proxy Adresse
Benutzer Auth



5.2- Verbindung Tests: Load-Balancing Server

```
web@cli4:~  
$>tracert round.webproxy.local  
tracert to round.webproxy.local (172.16.2.12), 30 hops max, 60 byte packets  
 1 192.168.3.1 (192.168.3.1)  0.132 ms  0.071 ms  0.238 ms  Imitation Server  
 2 172.16.2.12 (172.16.2.12)  0.527 ms  0.547 ms  0.763 ms  Proxy Adresse  
web@cli4:~  
$>curl round.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>  
web@cli4:~  
$>curl web:1234@round.webproxy.local  
apache 1  
web@cli4:~  
$>curl web:1234@round.webproxy.local  
apache 2  
web@cli4:~  
$>curl web:1234@round.webproxy.local  
apache 3  
web@cli4:~  
$>curl web:1234@round.webproxy.local  
apache 1  
web@cli4:~  
$>
```

Benutzer Auth

Round-Robin Load-Balancing



5.3- Proxy System Tests: Benutzer Einfügen

```
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
```

```
USERS MENU
1      Show Users
2      Add User
3      Update User
4      Delete User

Option: 1
Show Users...

web
test_user
```

```
root@proxy:~--#>cat /etc/nginx/htaccess/.htpasswd
web:$apr1$lhT.aNx8$5NvyD51LNJ391zg/N6I1z.
test user:$apr1$vGoeVlHq$dRiYcp8ji.SVlw9e3d5X91
```



5.4- Proxy System Tests: Single Server Einfügen

```
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: serversingle Name
New server name: serversingle.webproxy.local
Is that correct(Y/N): y
Please enter the IP address: 10.0.0.22 IP
New server IP: 10.0.0.22
Is that correct(Y/N): y
```

```
91 ###
92 ### ADDED
93 ###
94 auto eth1:added1 ← Adapter Name
95 iface eth1:added1 inet static
96         address 172.16.2.101 ← Adapter IP
97         netmask 255.255.255.0
"/etc/network/interfaces" 97 Zeilen --100%--
```



5.5- Proxy System Tests: Single Server Einfügen

```
root@proxy:~--#>cat /etc/bind/db.webproxy.local.forward | grep serversingle
serversingle IN A 172.16.2.101
root@proxy:~--#>cat /etc/bind/db.webproxy.local.rev | grep serversingle
101 IN PTR serversingle.webproxy.local
root@proxy:~--#>cat /etc/nginx/sites-available/serversingle
server {
    listen 172.16.2.101:80;
    server_name serversingle.webproxy.local;

    location / {
        proxy_pass http://10.0.0.22:80;
        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;
    }
}
root@proxy:~--#>ls /etc/nginx/sites-enabled/ | grep serversingle
serversingle
root@proxy:~--#>cat ~/scripts/firewall/servers/servers_singles | grep serversingle
serversingle_ip=172.16.2.101
root@proxy:~--#>cat ~/scripts/firewall/7_servers_singles.sh | grep serversingle
### BEGIN: serversingle
$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanB -d $serversingle_ip -j ACCEPT
$fw -A INPUT -p icmp --icmp-type 8 -i $to_server_adapter -s $lanC -d $serversingle_ip -j ACCEPT
$fw -A INPUT -p tcp --dport $port_http -i $to_server_adapter -s $lanB -d $serversingle_ip -j ACCEPT
$fw -A INPUT -p tcp --dport $port_http -i $to_server_adapter -s $lanC -d $serversingle_ip -j ACCEPT
### END: serversingle
```

DNS

Proxy(NginX)

Firewall



5.6- Proxy System Tests: Single Server Auflistung

SERVER MENU

```
1      Show Servers
2      Add Server
3      Delete Server
```

```
Option: 1
Show Servers...
```

```
SERVER_NAME: serversingle
            TYPE: single server
            IP: 172.16.2.101
            PROXY: 10.0.0.22
            ADAPTER: eth1:added1
```

```
root@proxy:~--#>cat ~/scripts/server_proxy/added_servers/serversingle
server_name=serversingle
type=single_server
new_server_ip=172.16.2.101
proxy_server_ip=10.0.0.22
adapter_name=eth1:added1
root@proxy:~--#>cat ~/scripts/server_proxy/variables/adapter_number
adapter_number=2
root@proxy:~--#>cat ~/scripts/server_proxy/variables/new_ip
server_ip=102
root@proxy:~--#>
```



5.7- Proxy System Tests: Single Server Löschen

```
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: serversingle Server Name
```

```
Server address: serversingle.webproxy.local
```

```
Is that correct(Y/N): y
```

```
root@proxy:~--#>cat /etc/network/interfaces | grep added ← Interface
root@proxy:~--#>cat /etc/bind/db.webproxy.local.forward | grep serversingle ← DNS
root@proxy:~--#>cat /etc/bind/db.webproxy.local.rev | grep serversingle ←
root@proxy:~--#>cat /etc/nginx/sites-available/serversingle ← Proxy(NginX)
cat: /etc/nginx/sites-available/serversingle: Datei oder Verzeichnis nicht gefunden
root@proxy:~--#>ls /etc/nginx/sites-enabled/ | grep serversingle ←
root@proxy:~--#>cat ~/scripts/firewall/servers/servers_singles | grep serversingle ← Firewall
root@proxy:~--#>cat ~/scripts/firewall/7_servers_singles.sh | grep serversingle ←
root@proxy:~--#>cat ~/scripts/server_proxy/added_servers/serversingle ← Config Datei
cat: /root/scripts/server_proxy/added_servers/serversingle: Datei oder Verzeichnis nicht gefunden
root@proxy:~--#>
```



5.8- Proxy System Tests: Load-Balancing Server Einfügen

```
ADD TO LOAD-BALANCING MENU

1      Add to Round-Robin
2      Add to Weighted
3      Add to Persistent
4      Add to Least Connected

Option: 1
Adding to Round-Robin...

Server to be added to: round.webproxy.local
Is that correct(Y/N): y
Please enter the IP address: 10.0.0.55 Server IP
New server IP: 10.0.0.55
Is that correct(Y/N): y
```



5.9- Proxy System Tests: Load-Balancing Server Einfügen

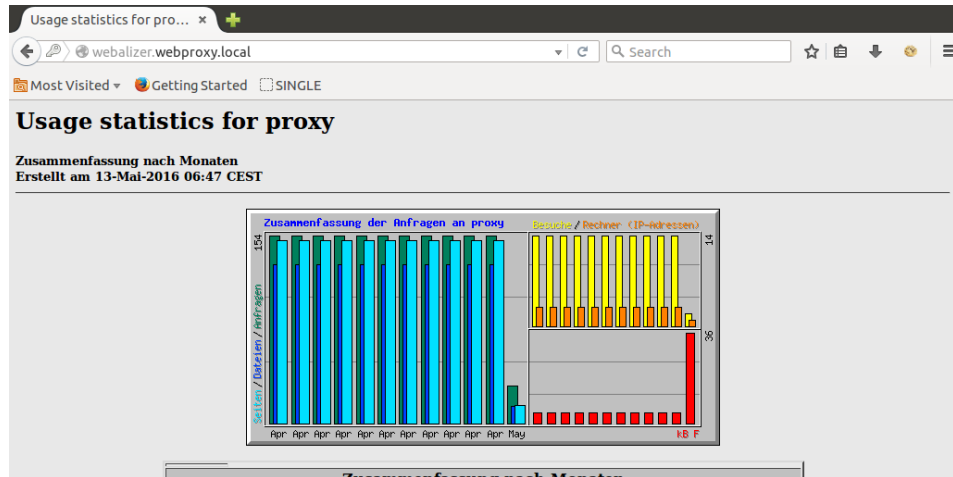
```
root@proxy:~--#>cat /etc/nginx/sites-available/round ← Proxy(NginX)
upstream web_round {
    server 10.0.0.11:80;
    #
    server 10.0.0.33:80;
    ### TO ADD
    server 10.0.0.55:80;
}

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;
    }
}

root@proxy:~--#>cat ~/scripts/server_proxy/added_servers/round_1 ← Config Datei
server_name=round_1
type=load-balancing
new_server_ip=10.0.0.55
proxy_server_ip=172.16.2.12
proxy_address=round.webproxy.local
root@proxy:~--#>cat ~/scripts/server_proxy/variables/balancing_counters | grep round_count ← Round-Robin LoadBalancing Counter
round_count=2
```



5.10- Log-Datei: WebAlizer



Usage statistics for pro... x

webalizer.webproxy.local/usage_201605.html

Most Visited Getting Started SINGLE

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

Einleitung

Install und Konfig

Tests

1

2

3

4

5

6

Planung

Implementierung

Abschluss



6.1- Qualitätsausforderung

- Benutzer Verbindung funktioniert
- Benutzer Authentifizierung funktioniert
- Load-Balancing funktioniert
- Einfügen/Auflisten/Löschen Server oder Container
 - Single funktioniert
 - Load-Balancing funktioniert
- Log-Datei bei WebAlizer Funktioniert

6.2- Erweiterbarkeit

- Modularessystem
- Skalierbarkeit



6.3- Fazit

- Lerneffekt
 - Netzwerkverständnis
 - Bashscripts
- Erfolgreich
 - Pilotprojekt abgeschlossen
- Problemen
 - Zeit/Komplexität unterschätzt
 - Sprache
- Ausblick
 - Pilotprojekt einsetzbar und erweiterbar

Quellenangabe

- Time is Money Bilder
 - <http://thecontextofthings.com/2015/04/12/time-is-literally-the-new-money/>
- Git Bilder
 - <http://kapitel26.github.io/slides/warum-wir-git-lieben/#/>
- Amazon Bilder
 - <http://www.parkmycloud.com/blog/love-aws-cloud-but-want-to-reduce-ec2-waste/>
- Azure Bilder
 - <http://wccftech.com/dev-confirms-power-microsofts-azure-cloud-explains-developers-reclutant/>
- SaaS Bilder
 - <http://talkincloud.com/saas-software-service/12012014/gartner-saas-deployments-are-now-mission-critical>
- Zeitablaufen Generierung
 - <https://www.meta-chart.com>

GitHub

<https://github.com/brunoreboucas>

Danke!