



## Loadbalancer in Web-basierten Anwendungen

Michael Hopf

München/HQ Bamberg Berlin Đà Nẵng Dresden Grenoble Hamburg Köln Leipzig Nürnberg Prag Washington Zug

Dresden, 18.5.2017

# Übersicht

1. Michael Hopf, MGM

2. Loadbalancer, Motivation, Ziele

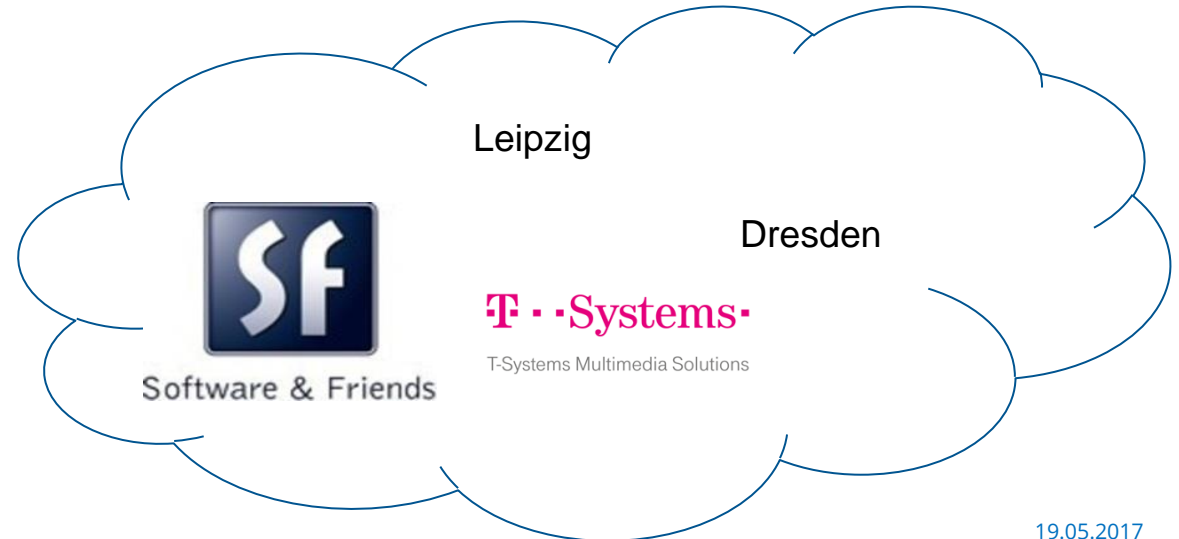
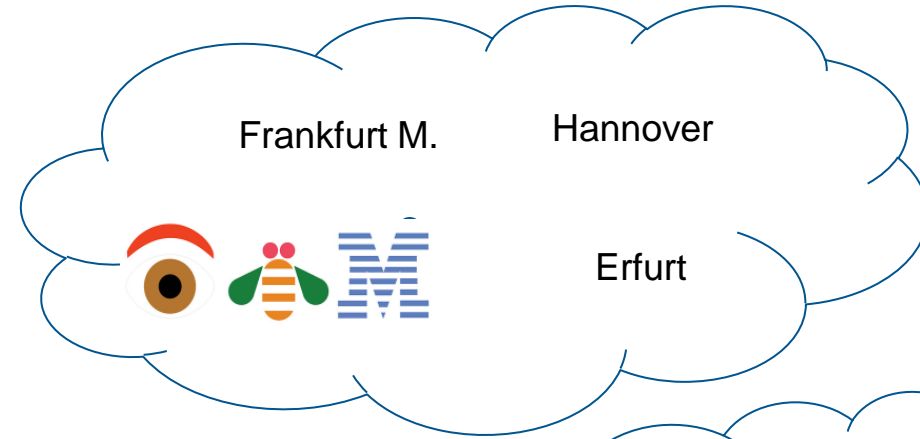
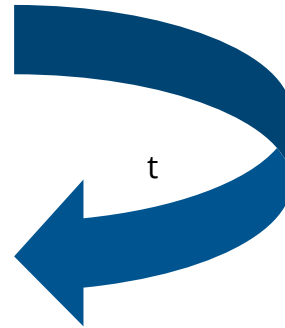
3. Loadbalancer Konzepte

4. Welche Anforderungen lassen sich mit Loadbalancermitteln erfüllen ?

5. Deployment Szenarien

6. Erfolgsfaktoren

# Michael Hopf



# MGM

- Softwarelösungshaus
- 500 Mitarbeiter
- 13 Standorte
- Hochskalierbare, integrierbare und sichere Online-Anwendungen



München



Bamberg



Berlin



Đà Nẵng



Dresden



Grenoble



Hamburg



Köln



Leipzig



Nürnberg



Prag



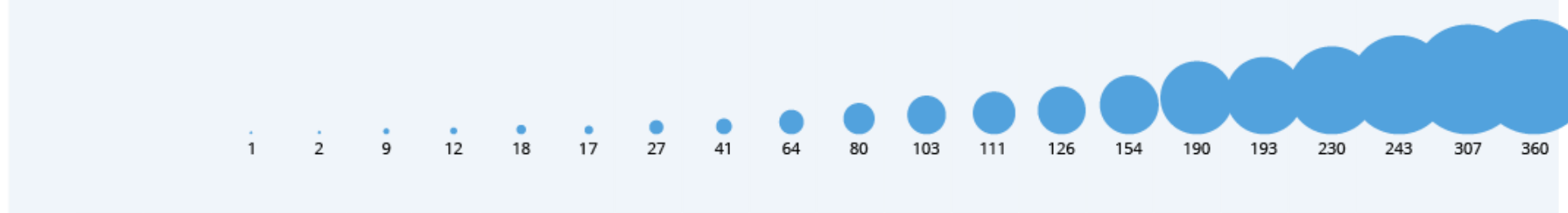
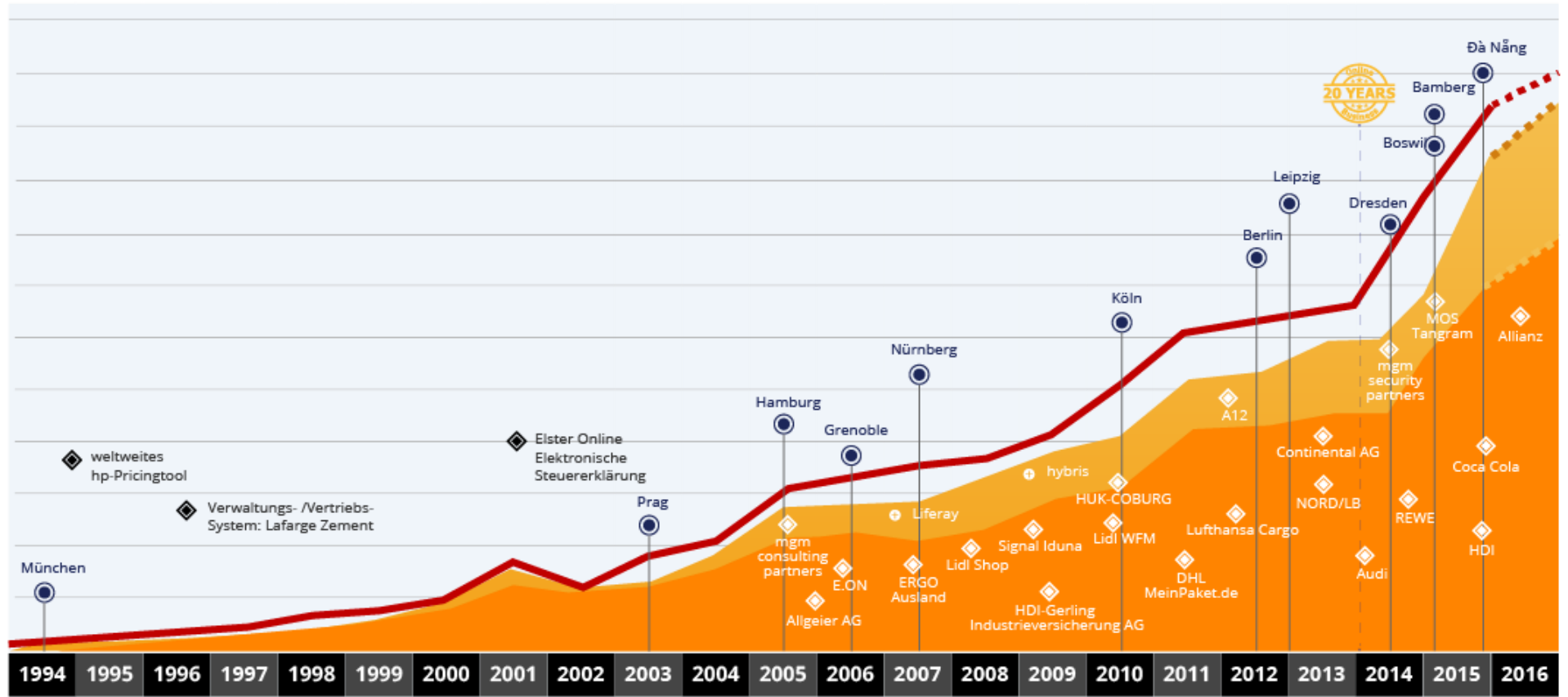
Washington



Zug

Mio. EUR

Mitarbeiter



- Umsatzentwicklung in Mio. EUR
- Meilensteine
- Partner
- Standorte
- Nationalität Deutsch
- Andere Nationen
- Anzahl Kinder

# Übersicht

1. Michael Hopf, MGM

2. Loadbalancer, Motivation, Ziele

3. Loadbalancer Konzepte

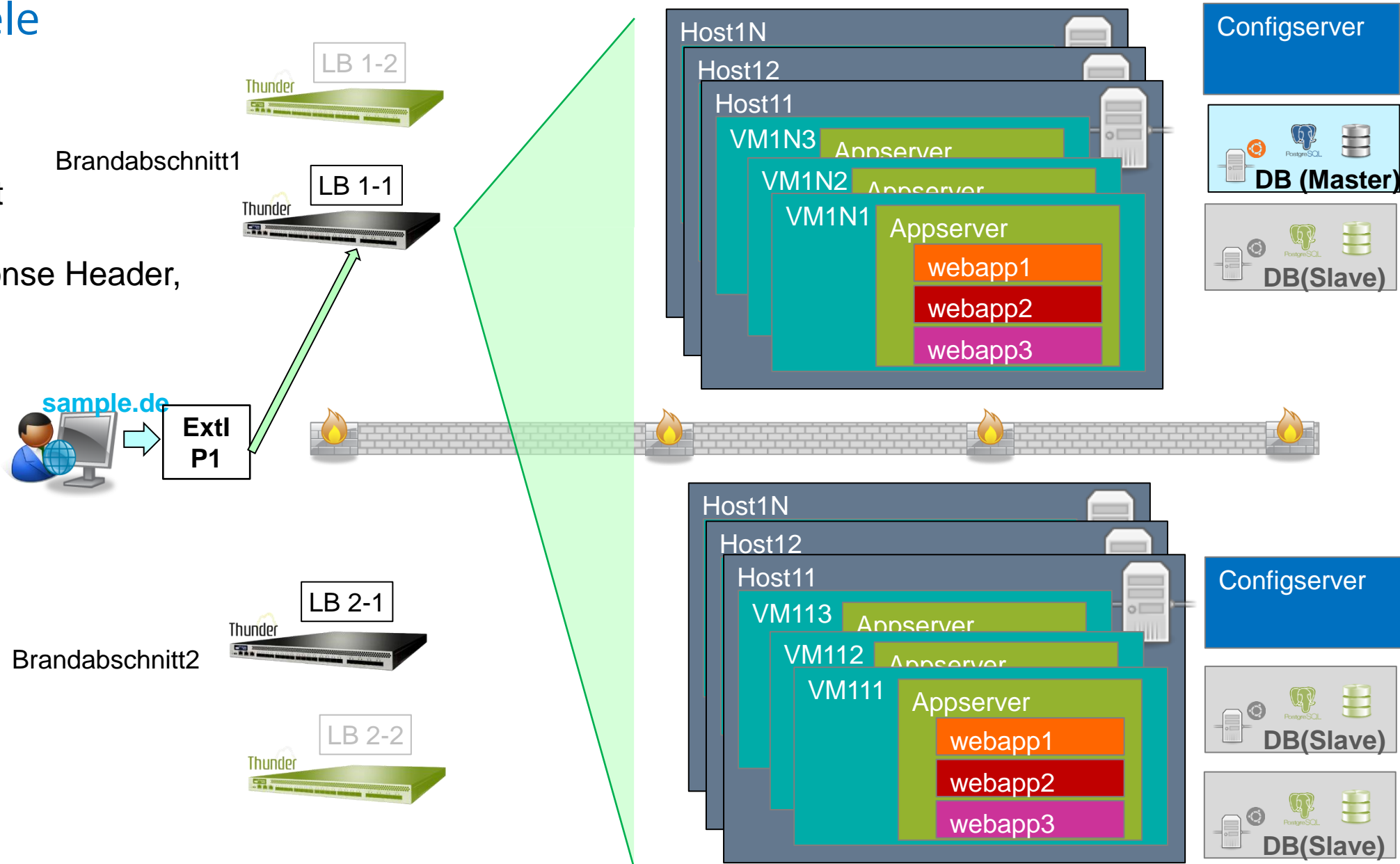
4. Welche Anforderungen lassen sich mit Loadbalancermitteln erfüllen ?

5. Deployment Szenarien

6. Erfolgsfaktoren

# Motivation, Ziele

- Performance
- Skalierbarkeit
- Hochverfügbarkeit
- Sicherheit (SSL, TLS, Response Header, ...)



# Übersicht

1. Michael Hopf, MGM

2. Loadbalancer, Motivation, Ziele

3. Loadbalancer Konzepte

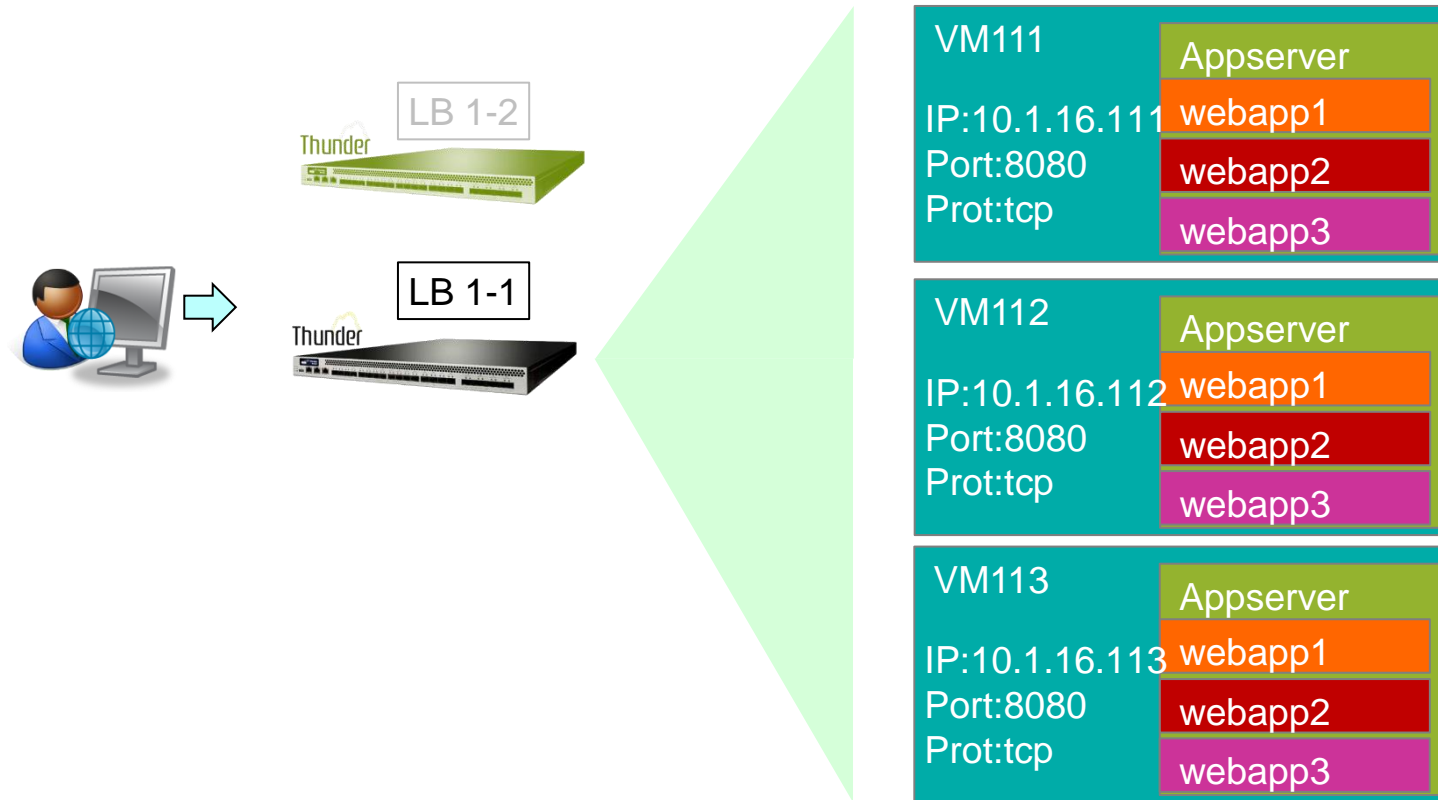
4. Welche Anforderungen lassen sich mit Loadbalancermitteln erfüllen ?

5. Deployment Szenarien

6. Erfolgsfaktoren



# Loadbalancer Konzepte – Real Server (IP+Ports+Protocols)



# Loadbalancer Konzepte – Real Server (IP+Ports+Protocols)



```
vThunder(config)# slb server rs111 10.1.16.111
vThunder(config-real server)# port 8080 tcp
```

Virtual Servers Virtual Services Service Groups Servers Policy Limits Class Lists Application Global Session vThunder 4.1.0-P8 build 98

ADC >> SLB >> Servers >> Update [Help](#)

### Update Server

Name *	rs111
Type	<input checked="" type="radio"/> IPv4 <input type="radio"/> IPv6 <input type="radio"/> FQDN
Host *	10.1.16.111
Action	Enable
Disable Health Check	<input type="checkbox"/>
Health Monitor	
Connection Limit	8000000
No Logging	<input type="checkbox"/>

Advanced Fields +

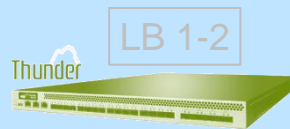
Port								
	Port	Protocol	Weight	Conn Limit	Health Check	Range	Conn Resume	Actions
<input type="checkbox"/>	8080	tcp	1	8000000	Default	0		Edit

Cancel Update

# Loadbalancer Konzepte – Virtual Server (VIP + Ports)



VIP: 192.168.4.4  
Port: 443  
Port: 80



sample.de



VM111	Appserver
IP:10.1.16.111	webapp1
Port:8080	webapp2
Prot:http	webapp3
VM112	Appserver
IP:10.1.16.112	webapp1
Port:8080	webapp2
Prot:http	webapp3
VM113	Appserver
IP:10.1.16.113	webapp1
Port:8080	webapp2
Prot:http	webapp3

# Loadbalancer Konzepte – Virtual Server (VIP + Ports)



```
vThunder(config)# slb virtual-server vs1 192.168.4.4
vThunder(config-real vserver)# port 443 https
vThunder(config-real vserver-port)#
```

ADC >> SLB >> Virtual Servers >> Update

[Help](#)

## Update Virtual Server

Name *	<input type="text" value="Portal"/>
Use-If-Ip	<input type="radio"/> Enable <input checked="" type="radio"/> Disable
Wildcard	<input type="checkbox"/>
Address Type *	<input checked="" type="radio"/> IPv4
IP Address *	<input type="text" value="192.168.4.4"/>
Netmask	<input type="text" value="/24"/>
Action	<input type="text" value="Enable"/>

Advanced Fields +

### Virtual Port

[Delete](#) [Create](#)

	Port Number	Port Range	Protocol	Actions
<input type="checkbox"/>	443	0	https	<a href="#">Edit</a>

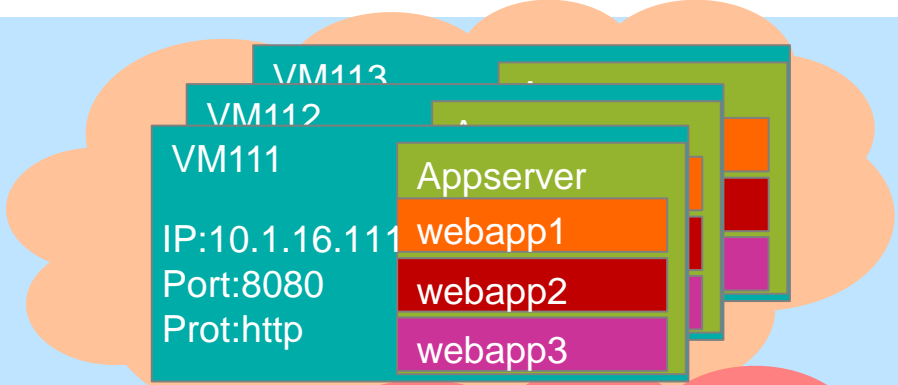
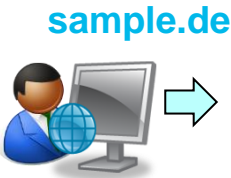
[Cancel](#)

[Update](#)

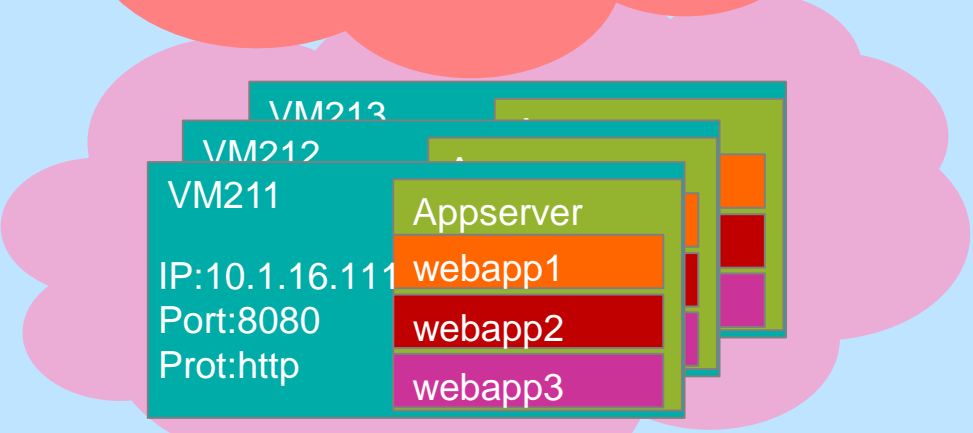
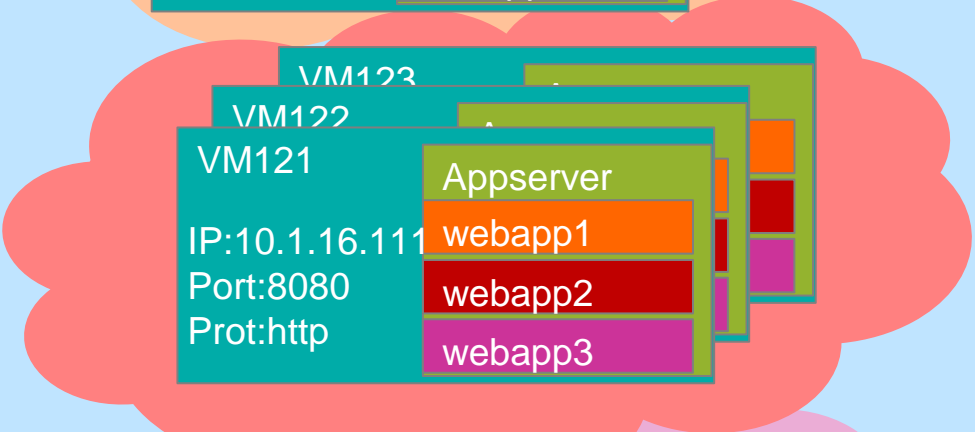
# Loadbalancer Konzepte – Service Groups (Pools)



VIP: 192.168.4.4  
Port: 443  
Port: 80



SG=(RealServer+Port)+  
SG:(RealServer+Port)=N:M



# Loadbalancer Konzepte – Service Groups (Pools)



```
vThunder(config)# slb service-group SG1 tcp
vThunder(config-real svc group)# method round robin
vThunder(config-real svc group)# member rs111 8080
vThunder(config-real svc group)# member rs112 8080
```

ADC >> SLB >> Service Groups >> Update

[? Help](#)

## Update Service Group

Name *	<input type="text" value="SG1"/>
Protocol	<input type="text" value="TCP"/>
Algorithm	<input type="text" value="Round Robin"/>
Health Check Disable	<input type="checkbox"/>
Health Monitor	<input type="text"/>

Advanced Fields +

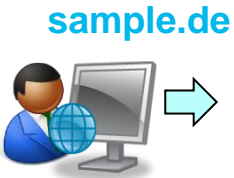
Member				
	Status	Name	Port	Actions
<input type="checkbox"/>		rs111	8080	<a href="#">Edit</a>
<input type="checkbox"/>		rs112	8080	<a href="#">Edit</a>

# Loadbalancer Konzepte – Health Monitors



- Überwachung der Verfügbarkeit von Real Servern
- Ping, Http, FTP, Pop, ...
- Zuordnung zu Real Servern (viele Real Server, viel Konfigurationsarbeit)
- Zuordnung Service Groups
  - Health Monitor gilt für alle Real Server der Service Group (spart Konfigurationsarbeit)
  - Bestimmt, ob Real Server in der Service Group verfügbar ist

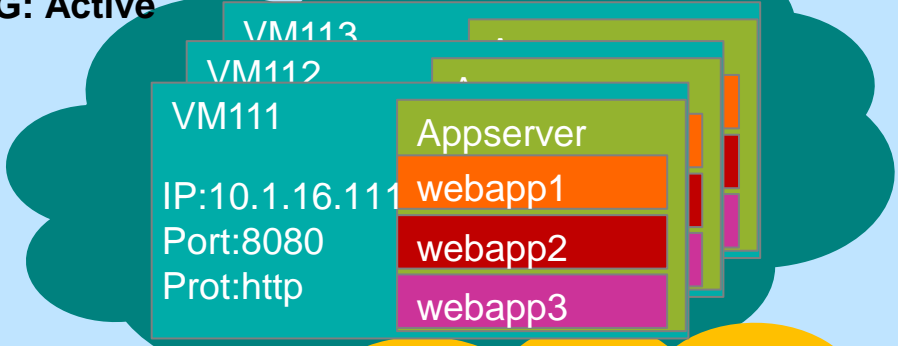
# Loadbalancer Konzepte - Health Monitors



VIP: 192.168.4.4  
Port: 443  
Port: 80

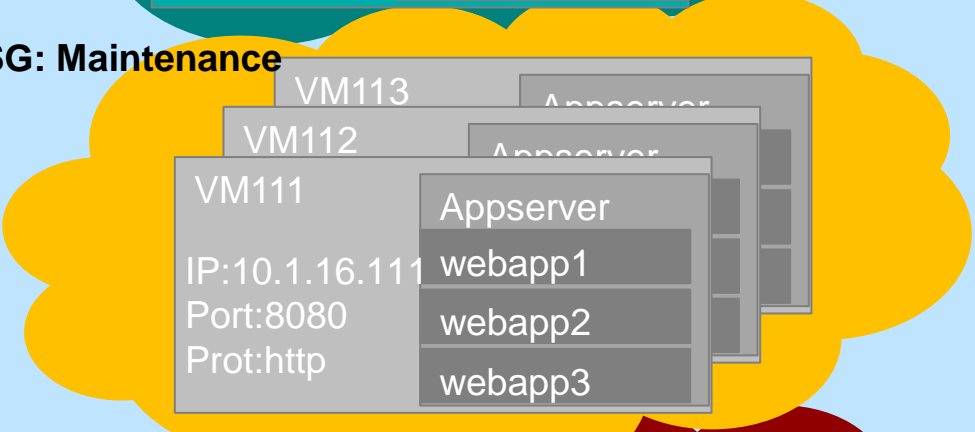


### SG: Active



Health Motior  
http\_active

### SG: Maintenance



Health Motior  
http\_maintenance

### SG: Test



Health Motior  
http\_test



# Loadbalancer Konzepte – Health Monitors



ADC >> Health Monitors >> Update

Help

## Create Health Monitors

General Fields	
Name *	<input type="text" value="http_active"/>
Method type	<input type="text" value="HTTP"/>
Retry	<input type="text" value="3"/>
Up Retry	<input type="text" value="1"/>
Interval	<input type="text" value="5"/>
Timeout	<input type="text" value="5"/>
Override Ipv4	<input type="text"/>
Override Ipv6	<input type="text"/>
Override Port	<input type="text"/>
SSL Cipher Suite	<input type="text" value="DEFAULT"/>
Passive	<input type="checkbox"/>
Strict Retry On	<input type="checkbox"/>
Server Err Resp	<input type="checkbox"/>

HTTP	
Specify URL string	<input checked="" type="checkbox"/>
URL Type	<input type="text" value="GET"/>
URL Path	<input type="text" value="/webapp1/lbstatus.jsp"/>
HTTP Host	<input type="text"/>
HTTP Port	<input type="text"/>
Maintenance Code	<input type="text"/>
HTTP Expect	<input type="text" value="HTTP Text"/>
HTTP Text	<input type="text" value="lbstatus_active"/>
Kerberos Auth	<input type="checkbox"/>
Username	<input type="text"/>
Password	<input type="text"/>

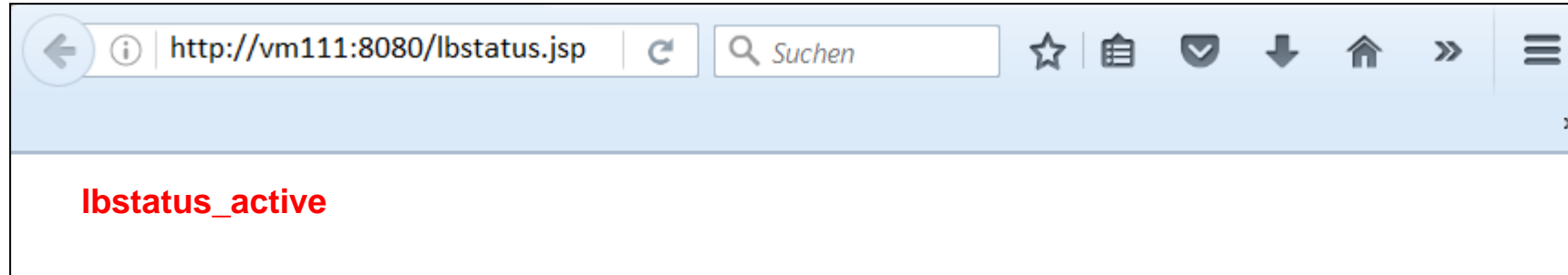
Cancel

Update Monitor

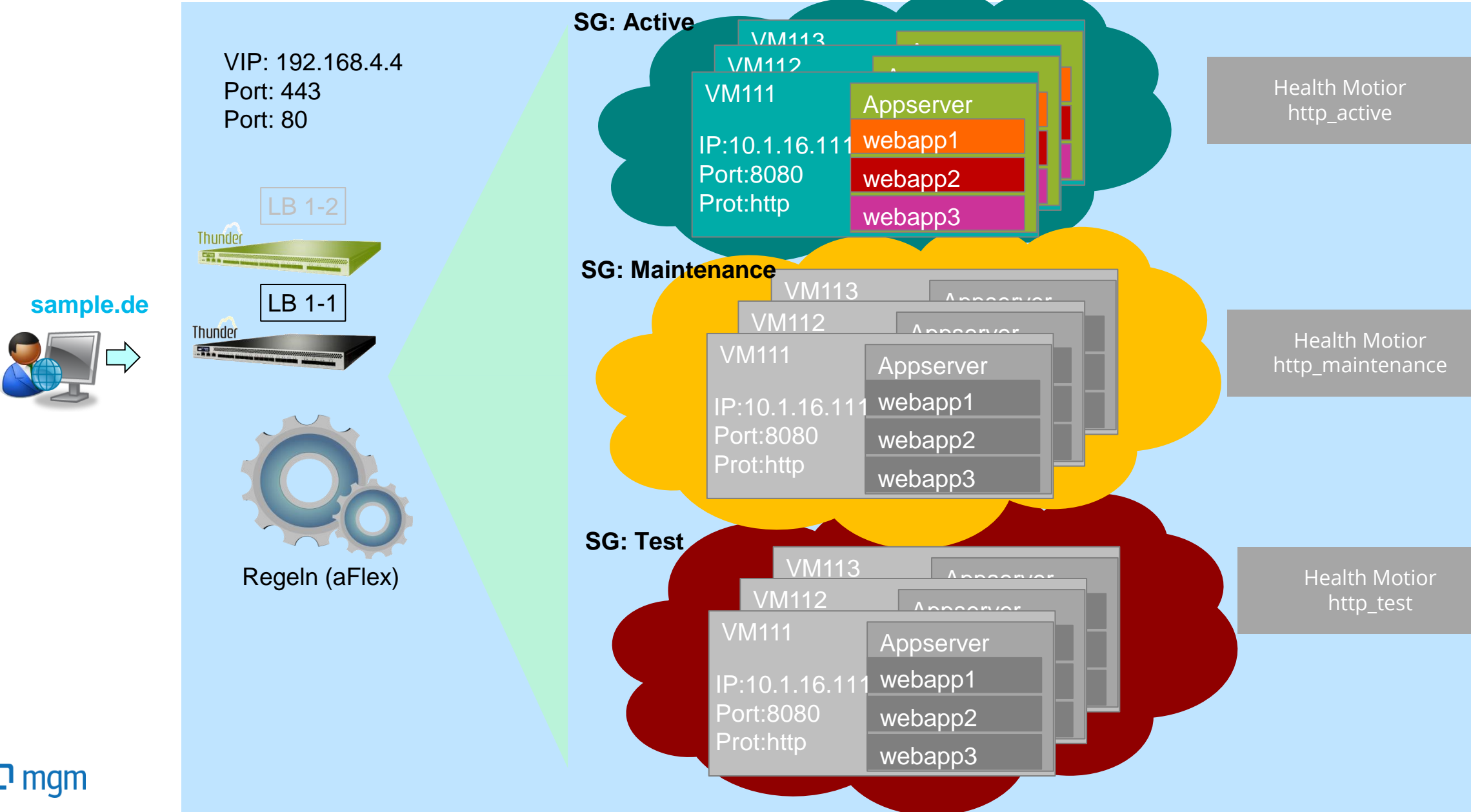
# Loadbalancer Konzepte – Health Monitors



- Webapps stellen Health Page bereit
- Mögliche Status: active, maintenance, inactive, test



# Loadbalancer Konzepte – Regel (aFlex)





## Regeln zur Request Verarbeitung

```
when HTTP_REQUEST {  
  if { [HTTP::uri] contains „lbstatus.jsp“ } {  
    HTTP::respond 403  
    event HTTP_REQUEST disable  
  }  
}
```



## Regeln zur Response Verarbeitung

```
when HTTP_RESPONSE {
  if { [HTTP::cookie exists JSESSIONID] } {
    if { ([HTTP::cookie exists "AddHttpOnly"]) } {
      HTTP::cookie value JSESSIONID "[HTTP::cookie value JSESSIONID];HttpOnly"
    }
    HTTP::cookie secure JSESSIONID enable
  }
  HTTP::cookie remove "AddHttpOnly"
  HTTP::header remove "Server"
  HTTP::header remove "Content-Location"
}
```



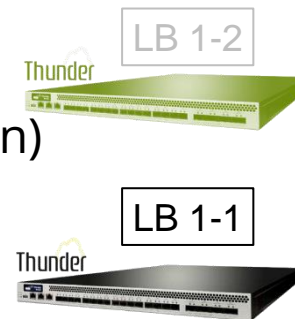
## Client Attribute abfragen, z.B. IP

```
if { [CLASS::match [IP::client_addr] testerips ip] } {  
    # Aufruf kommt von der IP eines Testers  
    # Prüfen, ob der Test-Pool verfügbar ist, dann selektieren  
    if { [active_members $testpool] > 0 } {  
        pool $testpool  
    }  
}
```

# Session Persistence



- Problem: Ein Nutzer hat eine Session auf einem Real Server. Wie kann sichergestellt werden, dass er seine Session nicht verliert, wenn der nächste Request auf einem anderen Real Server landet ?
- Lösung1: Sticky Session
  - IP basiert
  - Cookie basiert
    - Cookie vom Real Server (Probleme mit A10)
    - Cookie vom Loadbalancer
- Lösung2: Session Replication (zwischen den Appservern)



VM111	Appserver
IP:10.1.16.111	webapp1
Port:8080	webapp2
Prot:tcp	webapp3
VM112	Appserver
IP:10.1.16.112	webapp1
Port:8080	webapp2
Prot:tcp	webapp3
VM113	Appserver
IP:10.1.16.113	webapp1
Port:8080	webapp2
Prot:tcp	webapp3

# Loadbalancer Konzepte - Zusammenfassung



Virtual Server  
-IP  
-Port  
-Protocol

Service Group

Real Server  
- IP  
- Port  
- Protocol

HA/VRRP

Templates

aFlex Scripts (Rules)

SNat

Health Monitor

Persistence

Client SSL

Server SSL

Caching



# Übersicht

1. Michael Hopf, MGM

2. Loadbalancer, Motivation, Ziele

3. Loadbalancer Konzepte

4. Welche Anforderungen lassen sich mit Loadbalancermitteln erfüllen ?

5. Deployment Szenarien

6. Erfolgsfaktoren

## 5. Welche Anforderungen lassen sich mit Loadbalancermitteln erfüllen ?



- A0 HTTPS, SSL, TLS, CIPHER Suites
- A1 Response Location Header von HTTP auf HTTPS umschreiben (Redirects)
- A2 Client-IP-Adresse an den Servlet-Container weiterleiten
- A3 Sticky Sessions basierend auf JSessionId Cookie
- A4 Sonderbehandlungen für Login mit mobilen Endgeräten
- A5 Secure- und HttpOnly-Cookie
- A7 Redirect bei unbekanntem Context (<https://www.sample.de/gibtsjagarnicht/...>)
- A8 Schutz von bestimmten URLs (403)
- A9 Testzugang
- A10 Wartungsseite
- A11 Weiterleitung von Requests an bestimmte webapps an eigene Service Groups
- **A12 Unterstützung der Deployment Prozesse (Maintenance Mode)**
- ...

# Übersicht

1. Michael Hopf, MGM

2. Loadbalancer, Motivation, Ziele

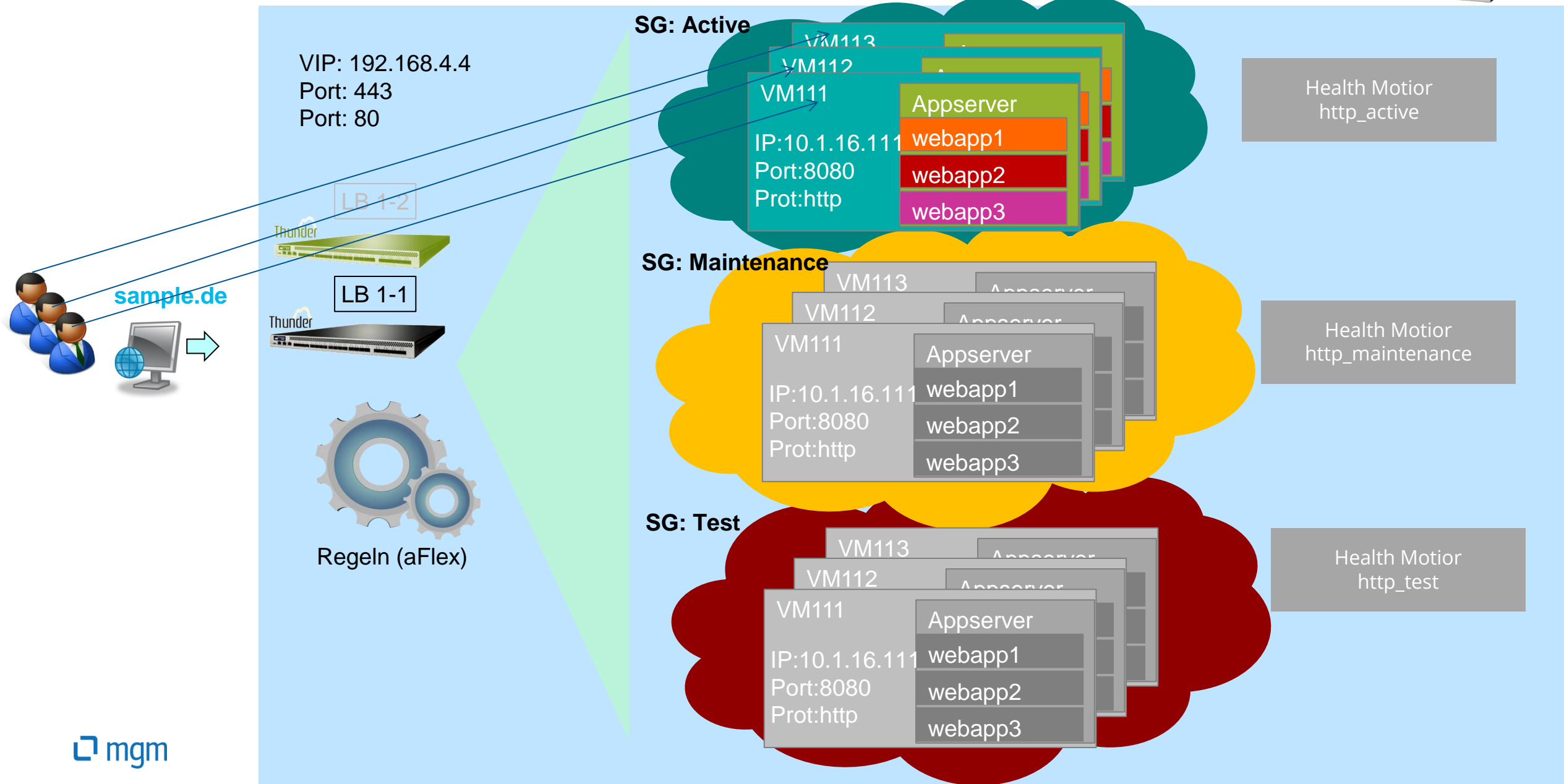
3. Loadbalancer Konzepte

4. Welche Anforderungen lassen sich mit Loadbalancermitteln erfüllen ?

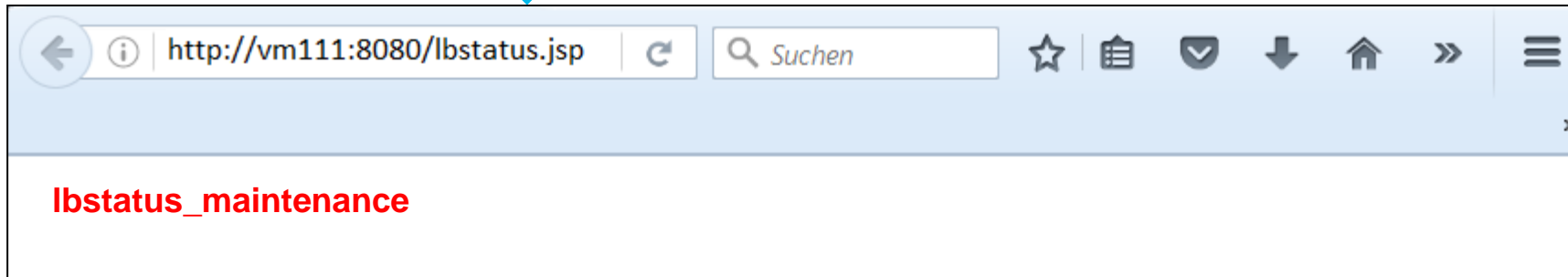
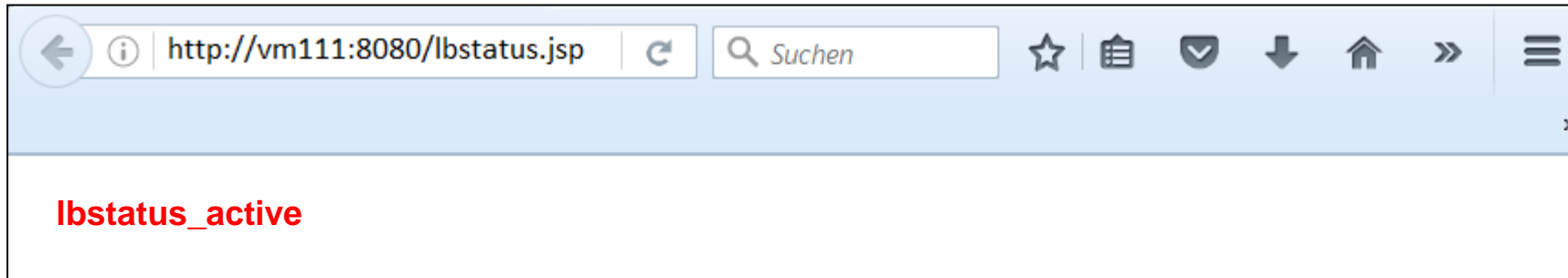
5. Deployment Szenarien

6. Erfolgsfaktoren

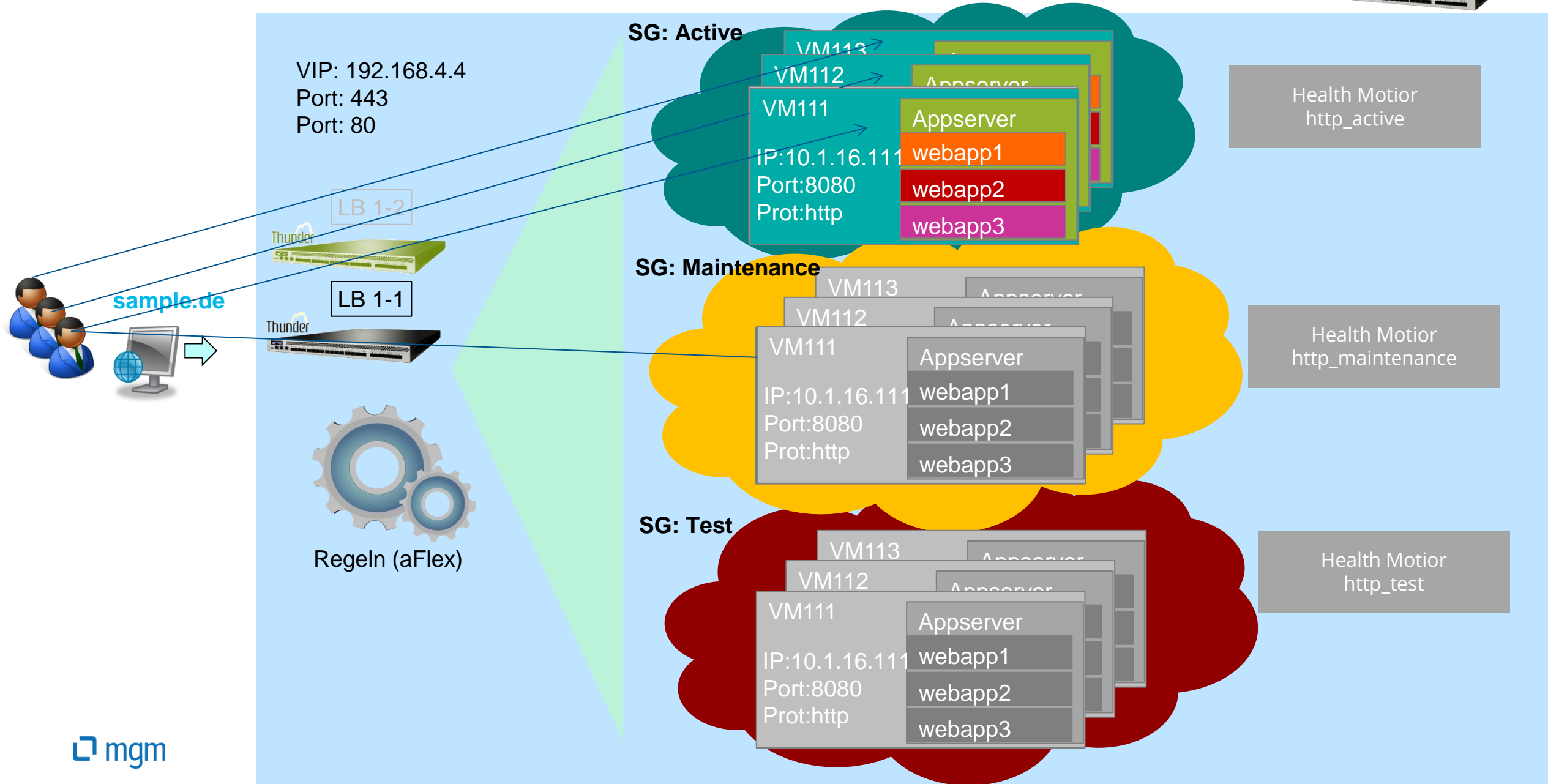
# Deployment Szenarien – Einen Real Server aktualisieren



# Deployment Szenarien – Einen Real Server aktualisieren



# Deployment Szenarien – Einen Real Server aktualisieren



# Deployment Szenarien – Einen Real Server aktualisieren



Browser address bar: <https://sample.de/form3.jsp> | Suchen

**Achtung, auf diesem Server werden um 15:00 Wartungsarbeiten durchgeführt.**  
Um auf einem anderen Server weiterarbeiten zu können, speichern Sie bitte Ihr Formular, loggen sich aus und schließen alle offenen Browser-Fenster. Starten Sie danach den Browser neu und melden Sie sich wieder an.

**ANGABEN ZU IHRER PERSON**

Adresse\*  Herr  Frau

Vorname\*

Nachname\*

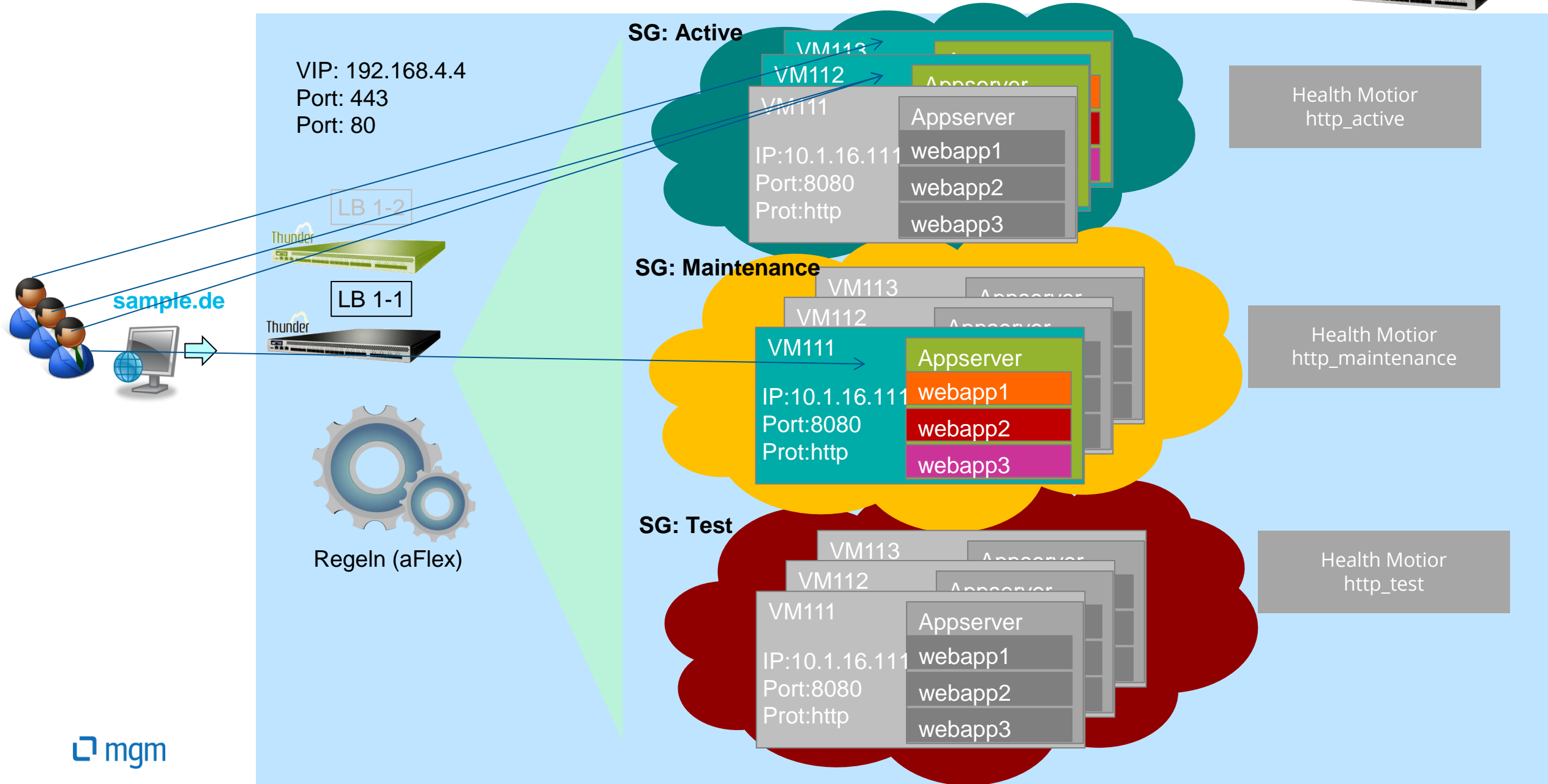
Geburtsname

Familienstand\*

Real Server ID  
Gut zu haben!

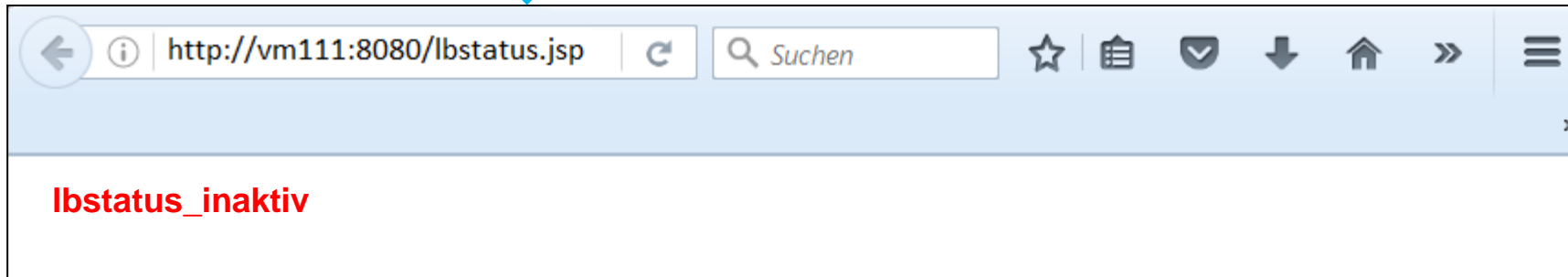
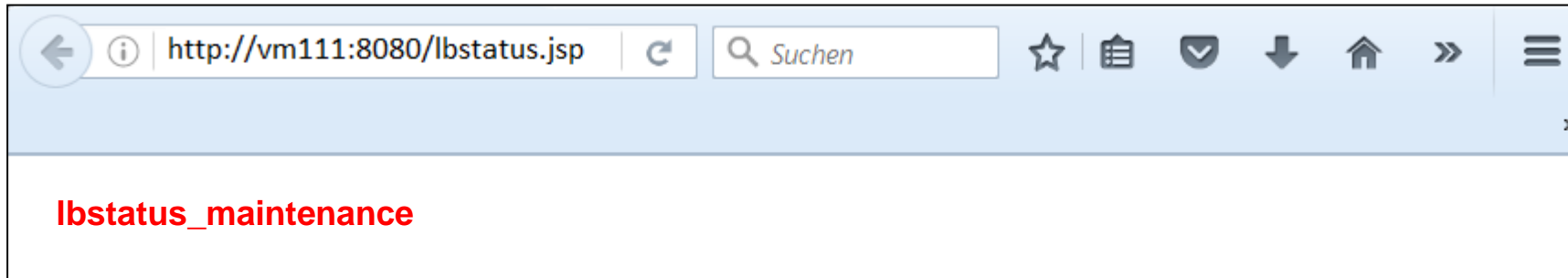
- 
- 
- VM111

# Deployment Szenarien – Einen Real Server aktualisieren

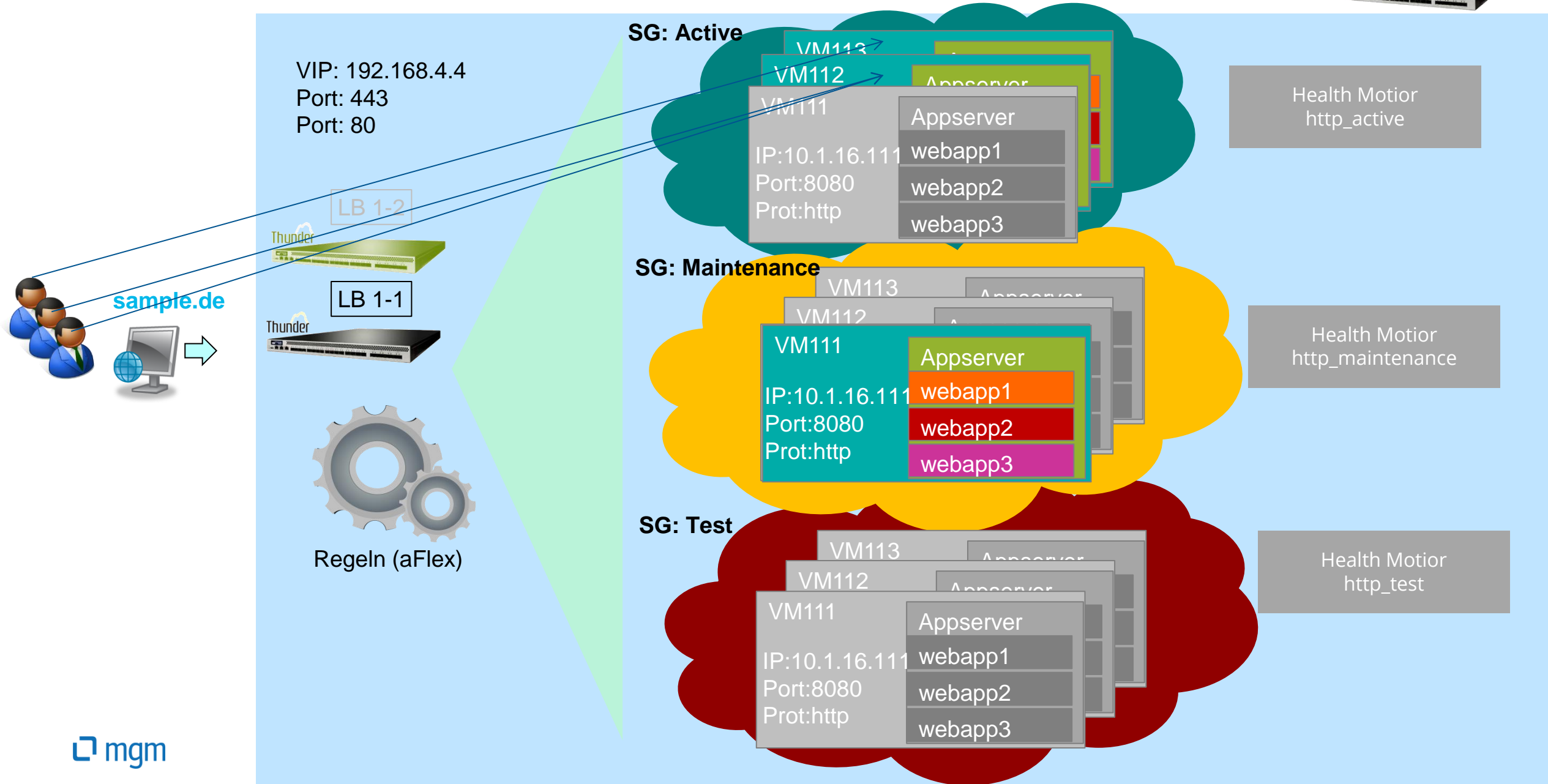




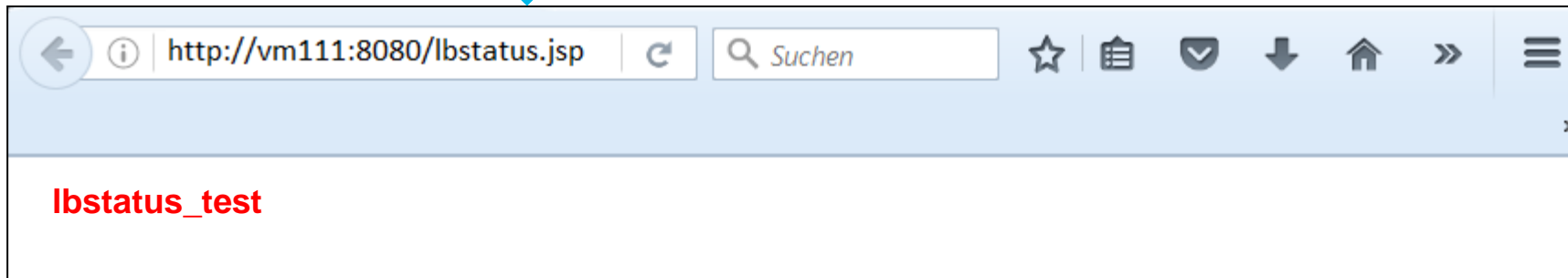
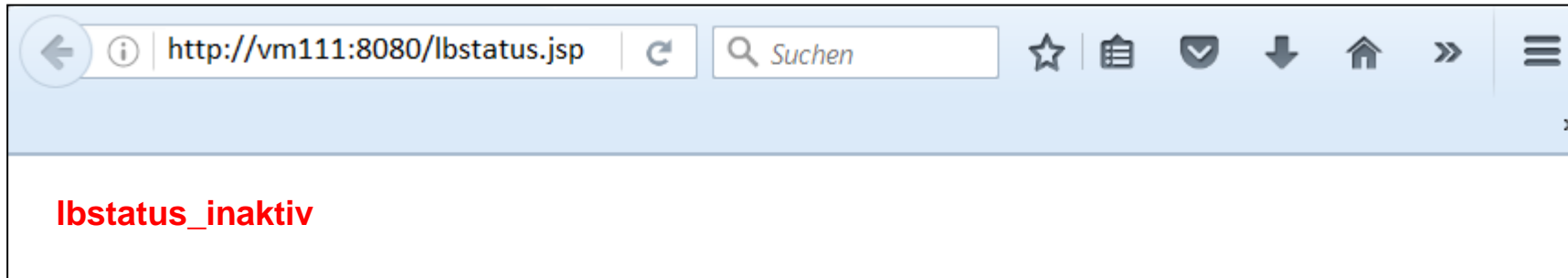
# Deployment Szenarien – Einen Real Server aktualisieren



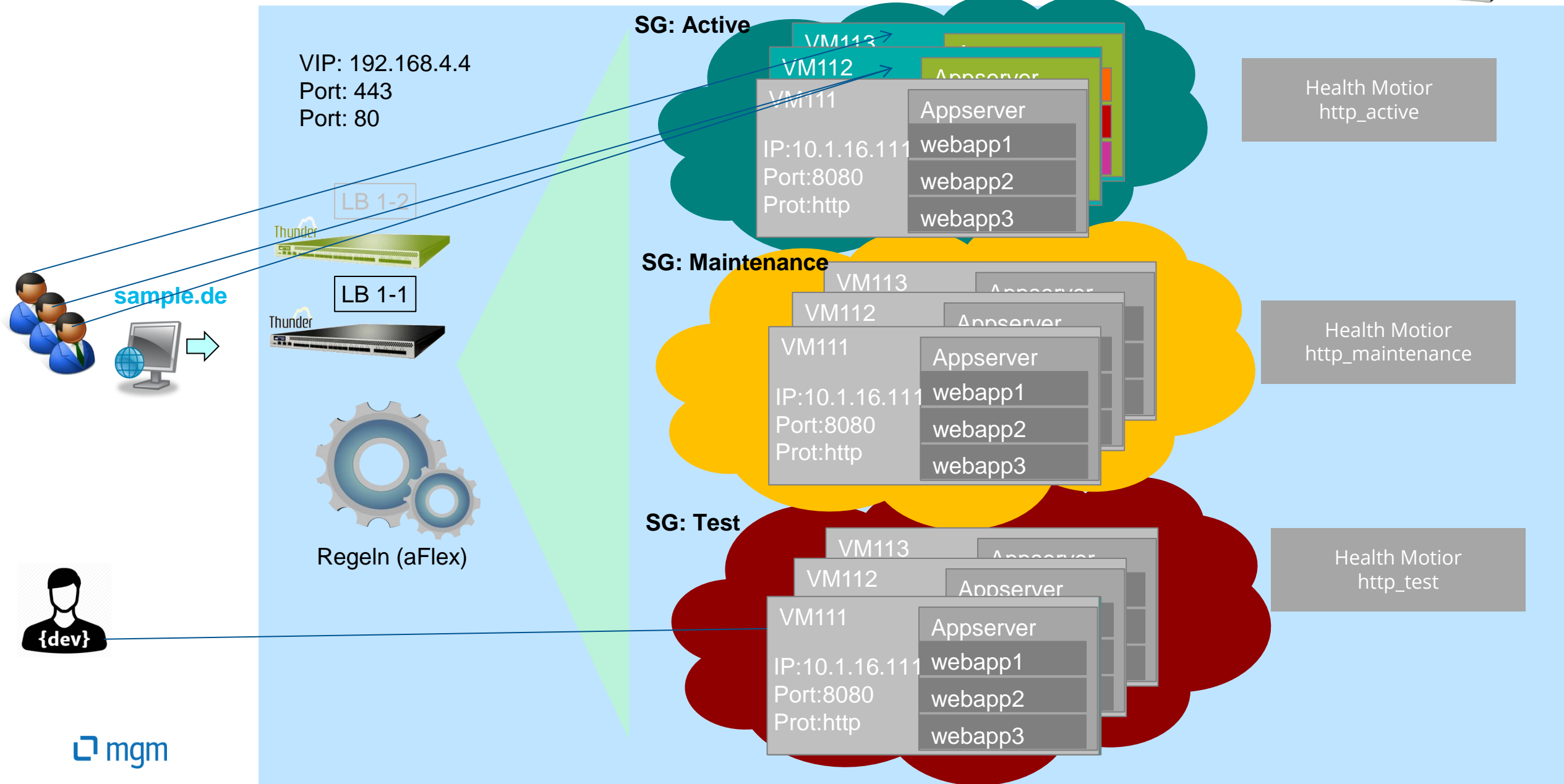
# Deployment Szenarien – Einen Real Server aktualisieren



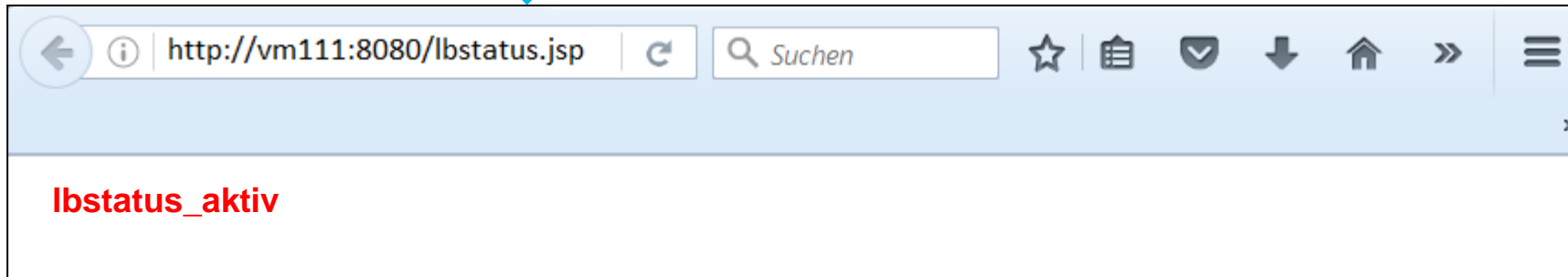
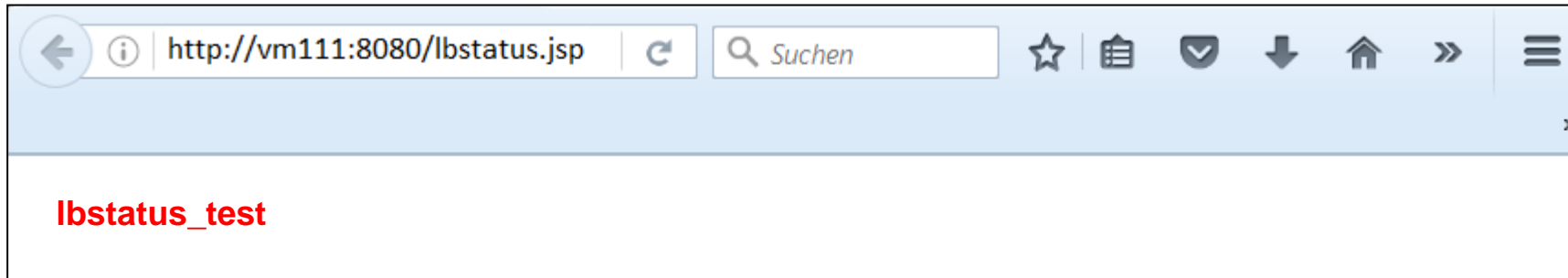
# Deployment Szenarien – Einen Real Server aktualisieren



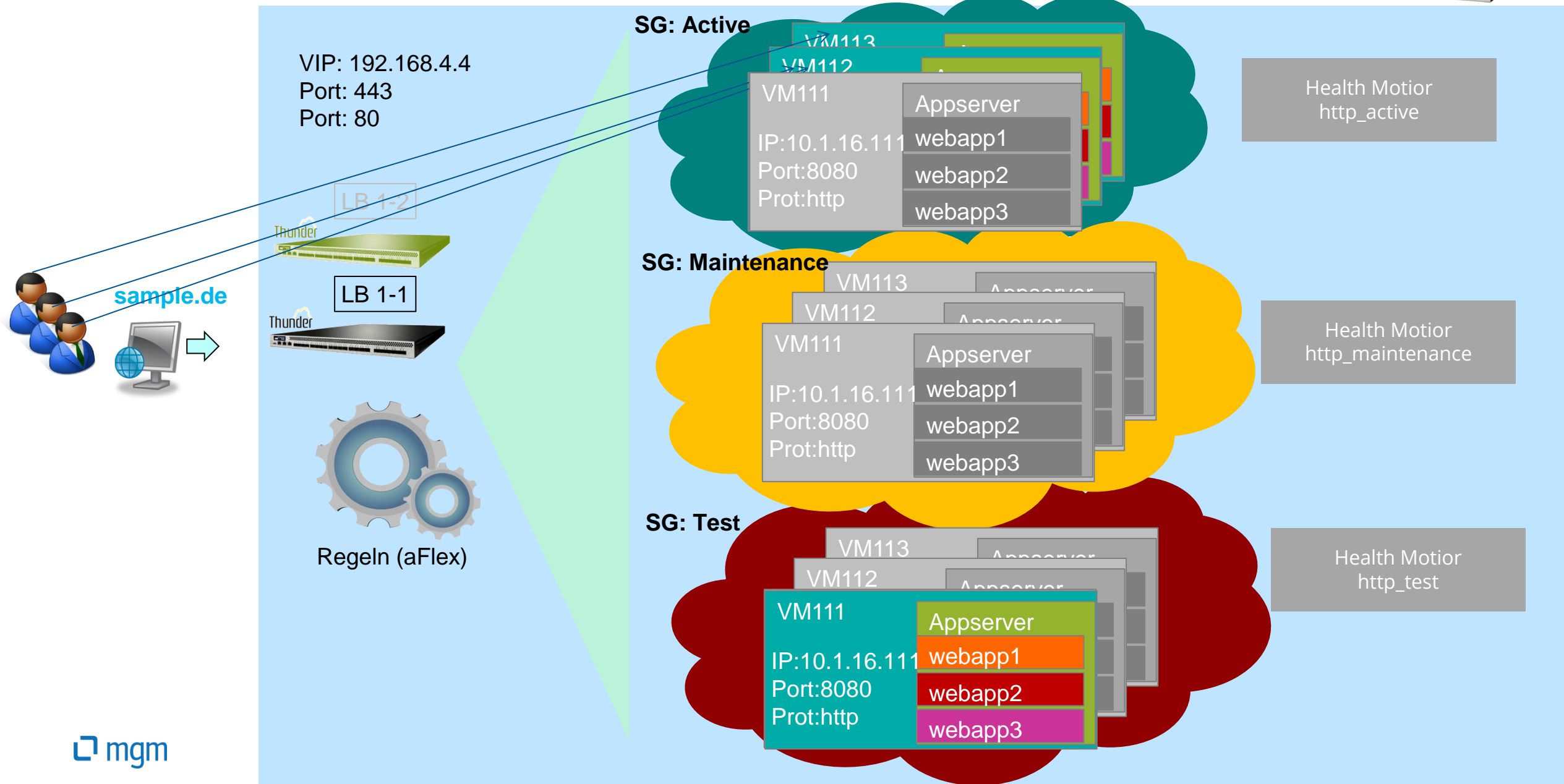
# Deployment Szenarien – Einen Real Server aktualisieren



# Deployment Szenarien – Einen Real Server aktualisieren



# Deployment Szenarien – Einen Real Server aktualisieren



# Übersicht

1. Michael Hopf, MGM

2. Loadbalancer, Motivation, Ziele

3. Loadbalancer Konzepte

4. Welche Anforderungen lassen sich mit Loadbalancermitteln erfüllen ?

5. Deployment Szenarien

6. Erfolgsfaktoren



- Es geht nur gemeinsam! Bewährt hat sich beim Thema Loadbalancer folgende Aufgabenteilung.
  - Betrieb ist verantwortlich für die Grundkonfiguration (Verkabelung, Ips, SNat, SSL Einrichtung und Zertifikatsverwaltung, Real Server, ...)
  - Anwendungsentwicklung steuert VMs, Pools, Regelwerk (aFlex), ... bei
- Installationsanleitungen: Skripte sind kompakter als Anweisung für Bedienung der Loadbalancer Admin Gui
- Änderungen am Loadbalancer können gravierende Auswirkungen auf die Verfügbarkeit der Anwendung haben. Deshalb sollten einmal eingerichtete Loadbalancer bei regulären Deployments der Anwedungen nicht angefasst werden! Das Maintenance Mode Feature ermöglicht das.



# Erfolgsfaktor - Entwicklungs- und Staging Umgebungen



- Die Entwicklung verfügt über Loadbalancer (VMs) in mindestens einem Entwicklungssystem.
- Im Integrationssystem (des Kunden) sind idealerweise dieselben Loadbalancer im Einsatz, wie im Produktivsystem.
- Aktualisierungen immer erst in Entwicklungs- und Staging-System testen, danach in Produktion bringen



- Komplette VMs als Liefergegenstand der Anwendungsentwicklung
- VMs (darauf installierte Anwendungen) sind frei von umgebungsspezifischer Konfiguration (Hostnamen, Ports, Nutzer, Passworte, Leistungsparameter, ...)
- Konfiguration wird erst bei Einspielung der VM in einer konkreten Zielumgebung zugesteuert
- Verteilung der Konfiguration über einen sog. Konfigurationsserver, VMs ziehen sich die Konfiguration vom Konfigurationsserver
- Konfiguration ist neu ladbar (ohne Neustart)
- Log- und Datenpartitionen auf separaten VM Images, die einen VM Tausch überdauern
- Einheitliche DNS Namen (mail, dbserver, logserver, configserver, ...) vereinfachen die Konfiguration
- Subdomains zur Unterscheidung der Umgebungen (dbserver.dev.sample.de, dbserver.test.sample.de, dbserver.production.sample.de, ...)
- Portschemata verwenden, um Trennung der Umgebungen sicherzustellen (20xyz-dev, 30xyz-test, 40xyz-integration, 50xyz-production)
- ...

# Erfolgsfaktor – schnelle, automatisierte Builds und Deployments



- Zeit für das Bauen einer Auslieferung und Deployment optimieren
- automatisierte Deployments



# Erfolgsfaktor – Trainings und Support



- Trainings der Loadbalancer Hersteller (oder Ihrer Partner) zu besuchen ist teuer, aber dennoch gut investiertes Geld
- Supportverträge



Vielen Dank!

# Innovation Implemented.



München



Bamberg



Berlin



Đà Nẵng



Dresden



Grenoble



Hamburg



Köln



Leipzig



Nürnberg



Prag



Washington



Zug

## **mgm technology partners GmbH**

Frankfurter Ring 105a

80807 München

Tel.: +49 (89) 35 86 80-0

Fax: +49 (89) 35 86 80-288

<http://www.mgm-tp.com>