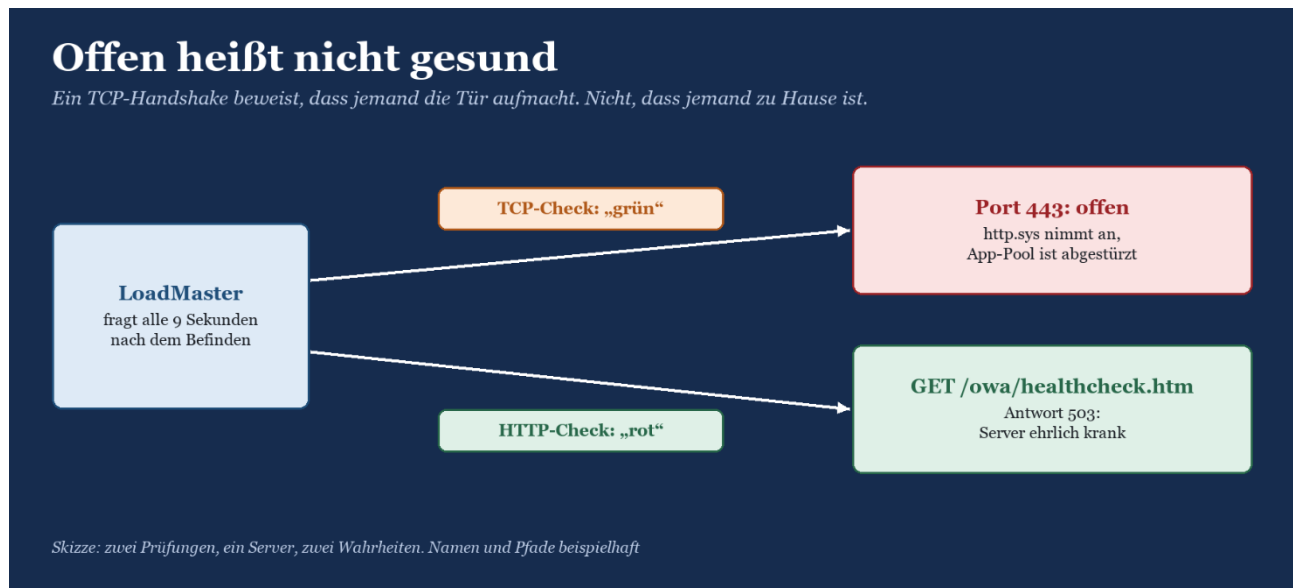


Health Checks, die wirklich prüfen: warum „Port 443 offen“ nicht reicht



WISSEN

Alles zum Kemp LoadMaster mit Microsoft-Brille: Exchange, SharePoint, ADFS, RDS und Kerberos an einem Ort.

› [Kemp LoadMaster](#)

BERATUNG

Deine Health Checks melden Grün, die Benutzer melden Rot? Wir prüfen gemeinsam, was deine Checks wirklich prüfen.

› [Beratung zum Kemp LoadMaster](#)

SCHULUNG

Grundlagen und Praxis für dein Admin-Team: vom ersten Virtual Service bis zum Health Check, der den Namen verdient.

› [Kemp-LoadMaster-Schulung](#)

Es gibt einen Satz, der in IT-Abteilungen mehr Wochenenden ruiniert hat als jeder Hardwaredefekt: „Der Load Balancer zeigt alles grün.“ Er fällt typischerweise um 7:40 Uhr an einem Montag, während die Hälfte der Belegschaft in OWA eine weiße Seite anstarrt und die andere Hälfte fröhlich Mails schreibt. Der LoadMaster lügt dabei nicht. Er beantwortet exakt die Frage, die du ihm gestellt hast. Und die lautete leider: „Nimmt auf Port 443 irgendetwas den Hörer ab?“

Ein offener Port ist ungefähr so aussagekräftig wie ein Puls, den man an einem Patienten misst, der gerade an einer Herz-Lungen-Maschine hängt. Technisch lebt da etwas. Ob es Mails zustellen, Dokumente öffnen oder Tokens ausstellen kann, steht auf einem anderen Blatt. Ein kemp health check ist genau dann etwas wert, wenn er die Frage stellt, die deine Benutzer stellen: „Funktioniert der Dienst?“ Nicht: „Läuft der Listener?“

In diesem Beitrag geht es um den Unterschied zwischen TCP- und HTTP(S)-Prüfungen, um Check-URL, erwartete Statuscodes, Host-Header, Intervalle und Retries. Danach wird es konkret: Exchange mit healthcheck.htm je Protokoll, SharePoint mit seiner Anmeldepflicht und ADFS mit /adfs/probe. Und zum Schluss der Klassiker, den du garantiert schon gesehen hast: Der Check schlägt fehl, jemand stellt „vorübergehend“ auf TCP um, und ein halbtoter Server bekommt wieder fröhlich Verkehr. Die Einordnung in die ganze Serie, von Grundlagen bis Kerberos-Delegation, findest du im [Kemp LoadMaster: der Wissensbereich](#).

FAKTEN • Health Checks am LoadMaster in Kürze

Der LoadMaster prüft jeden Real Server eines Virtual Service oder SubVS in einem festen Takt. Schlägt die Prüfung fehl, nimmt er den Server aus der Verteilung, bis die Prüfung wieder gelingt. Prüftypen reichen von ICMP und reiner TCP-Verbindung über protokollspezifische Checks (etwa SMTP oder LDAP) bis zu HTTP und HTTPS mit frei wählbarer URL, Methode, Statuscodes, Host-Header und Mustervergleich im Antworttext.

Standardwerte der Service-Checks laut Herstellerdokumentation: Check Interval 9 Sekunden, Connect Timeout 4 Sekunden, Retry Count 2. Seit Firmware 7.2.52 lassen sich diese Werte auch je Virtual Service oder SubVS setzen, vorher galten sie nur global.

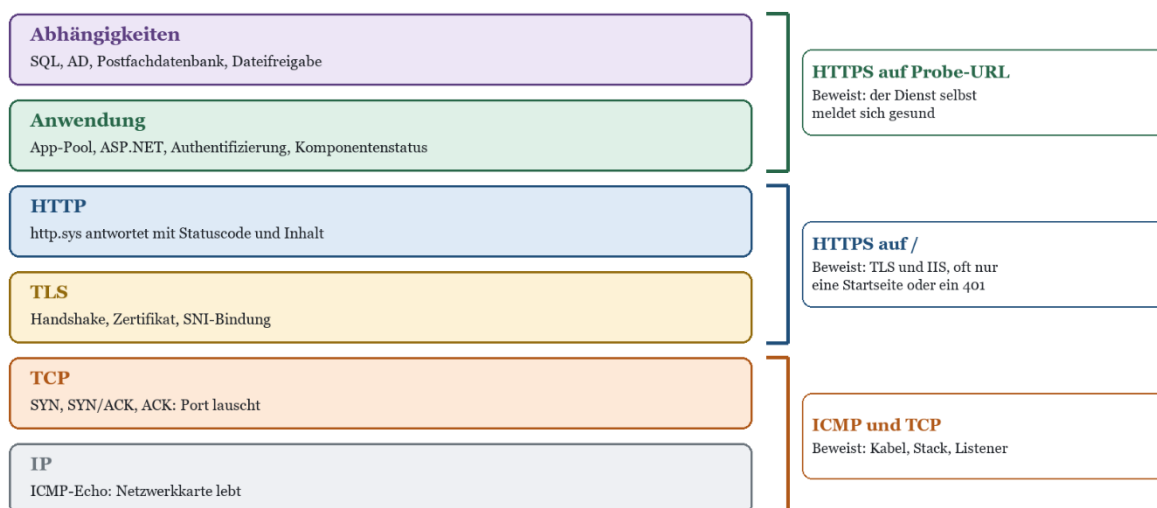
Ohne eigene Vorgabe wertet der HTTP(S)-Check die Statuscodes 200 bis 299, 301, 302 und 401 als „Up“.

Was ein Health Check beweist und was nicht

Bevor du an Stellschrauben drehst, lohnt sich ein ehrlicher Blick darauf, was eine Prüfung überhaupt beweisen kann. Jede Prüfung arbeitet auf einer bestimmten Schicht, und sie beweist nur, dass alles bis zu dieser Schicht funktioniert. Alles darüber ist Hoffnung.

Was welche Prüfung wirklich sieht

Je weiter oben die Antwort entsteht, desto mehr sagt sie über den Dienst aus



Skizze: Prüftiefe von ICMP bis zur Probe-URL, schematisch

Der TCP-Check: Puls messen am Türrahmen

Beim TCP-Check baut der LoadMaster eine Verbindung zum Real Server auf. Kommt der Handshake zustande, ist der Server „Up“. Das ist schnell, billig und für Dienste ohne HTTP oft das Einzige, was sinnvoll geht. Für Webdienste unter Windows ist es aber nahezu wertlos, und das hat einen ganz konkreten Grund: Auf einem IIS nimmt nicht die Anwendung die Verbindung an, sondern http.sys, der HTTP-Treiber im Kernel. Der lauscht auf Port 443, solange der Dienst „World Wide Web Publishing“ läuft und eine Bindung existiert. Ob der Anwendungspool dahinter gestoppt ist, im Rapid-Fail-Schutz hängt oder seine Datenbank nicht findet, interessiert http.sys beim Handshake nicht im Geringsten.

Das Ergebnis kennst du: Der App-Pool ist nach fünf Abstürzen in Folge deaktiviert, IIS liefert brav HTTP 503 „Service Unavailable“, und der TCP-Check meldet einen kerngesunden Server. Ein Bestatter, der den Puls am Türrahmen misst, kommt zu ähnlich optimistischen Befunden.

Der HTTP(S)-Check: eine echte Frage stellen

Der HTTP- oder HTTPS-Check geht weiter. Der LoadMaster baut die Verbindung auf, beim HTTPS-Check inklusive TLS-Handshake, schickt eine HTTP-Anfrage an eine von dir festgelegte URL und bewertet die Antwort. Standardmäßig ist das eine HEAD-Anfrage nach HTTP/1.0 auf den Pfad „/“. Optional kannst du GET, POST oder OPTIONS verwenden, HTTP/1.1 mit eigenem Host-Header schicken, bis zu vier eigene Header mitgeben und bei einer 200-Antwort den Inhalt gegen einen regulären Ausdruck prüfen.

Damit beweist der Check deutlich mehr: Die TLS-Bindung funktioniert, IIS verarbeitet Anfragen, und die Anwendung liefert auf genau dieser URL eine Antwort, die du als gesund definiert hast. Wie viel das wert ist, hängt vollständig von der URL ab. Die Startseite „/“ eines Exchange-Servers liefert eine Weiterleitung, die Standardwerte zählen 302 als gesund, und schon prüfst du mit großem Aufwand dasselbe wie der TCP-Check, nur mit TLS obendrauf.

Prüftypen im Vergleich

Prüfung	Was sie beweist	Was sie übersieht	Sinnvoll für
ICMP	Netzwerkkarte und IP-Stack antworten	Alles, was einen Dienst ausmacht	Fast nichts, allenfalls Infrastruktur ohne Dienst-Port
TCP	Ein Prozess lauscht auf dem Port	Abgestürzte App-Pools, 503, falsche Bindung, kaputte Abhängigkeiten	RDP, proprietäre Protokolle, Dienste ohne Prüfmöglichkeit
Protokoll-Check (SMTP, LDAP u. a.)	Der Dienst spricht sein Protokoll	Fachliche Fehler hinter dem Banner	SMTP-Connectoren, LDAP-Proxies
HTTP(S) auf „/“	TLS und IIS antworten	Fast alles, wenn „/“ nur weiterleitet oder 401 liefert	Einfache statische Seiten
HTTP(S) auf Probe-URL	Der Dienst bewertet sich selbst	Nur das, was die Probe nicht prüft	Exchange, ADFS, Eigenentwicklungen mit Health-Endpoint

WARNUNG · Die Standard-Statuscodes sind großzügiger, als du denkst

Ohne eigene Liste gelten 200 bis 299, 301, 302 und 401 als gesund. Das heißt: Eine Weiterleitung auf eine Fehlerseite ist „Up“. Eine Anmeldeaufforderung ist „Up“. Ein Server, dessen Anwendung nur noch auf eine Wartungsseite umleitet, ist ebenfalls „Up“. Wenn du nur das Protokoll von TCP auf HTTPS umstellst und den Rest auf Standard lässt, hast du häufig wenig gewonnen.

Die Stellschrauben: URL, Statuscodes, Host-Header, Methode

Die eigentliche Arbeit steckt in vier Einstellungen. Wer sie bewusst setzt, bekommt einen Check, der etwas aussagt. Wer sie auf Standard lässt, bekommt einen Check, der vor allem gut aussieht.

Die Check-URL: frag nach etwas, das kaputtgehen kann

Die URL trägst du nur als Pfad ein, also ohne Protokoll und ohne Hostnamen, zum Beispiel /owa/healthcheck.htm. Der Pfad sollte drei Bedingungen erfüllen. Erstens muss er anonym erreichbar sein oder du musst bewusst mit der Anmeldung umgehen, dazu gleich mehr bei SharePoint. Zweitens muss er von genau der Anwendung beantwortet werden, um die es geht, also nicht von einer statischen Datei im Stammverzeichnis der Standardwebsite. Drittens sollte er billig sein: Der LoadMaster fragt alle paar Sekunden, je Real Server und je SubVS. Bei fünf Exchange-

Protokollen und vier Servern sind das schon zwanzig Prüfungen pro Takt. Eine Probe-Seite, die bei jedem Aufruf eine Volltextsuche anstößt, ist ein Lasttest mit Ansage.

Erwartete Statuscodes: was „gesund“ heißt, legst du fest

Sobald du eine eigene Liste von Statuscodes hinterlegst, ersetzt sie die Standardliste. Zwei Eigenheiten musst du dabei kennen: Codes aus dem Bereich 200 bis 299 gelten immer als gesund, egal was in der Liste steht, und eintragen kannst du nur Codes zwischen 300 und 599. Du kannst also nicht sagen „nur 200, aber kein 204“. Du kannst aber sehr wohl sagen „401 ist kein gesund mehr“, indem du eine Liste setzt, die 401 nicht enthält. Umgekehrt kannst du 401 ausdrücklich als gesund erlauben und 301 und 302 weglassen.

Antwort	Standardbewertung	Was sie in der Praxis meist bedeutet	Empfehlung
200	gesund	Seite wurde ausgeliefert	Für Probe-URLs der einzige Code, den du willst
301, 302	gesund	Weiterleitung, oft auf Anmeldeseite oder https	Für Probe-URLs streichen
401	gesund	Windows-Anmeldung gefordert, IIS und Auth-Modul leben	Nur bewusst als Notlösung
403	krank	Zugriff verweigert, oft IP-Beschränkung oder SSL-Einstellung	Ursache finden, nicht erlauben
404	krank	Pfad falsch, Bindung falsch, Host-Header passt nicht	Check-URL und Host prüfen
500	krank	Anwendung wirft Ausnahme	Genau dafür gibt es den Check
503	krank	App-Pool gestoppt oder Komponente bewusst offline	Genau dafür gibt es den Check

Host-Header: der IIS will wissen, wen du meinst

Hier knallt es in Microsoft-Umgebungen am häufigsten. Der Standard-Check spricht HTTP/1.0 und schickt keinen sinnvollen Host-Header mit. Hat dein IIS nur eine Website mit einer Bindung auf alle Adressen, ist das egal. Hängen aber mehrere Websites mit Hostnamen-Bindungen auf demselben Port, entscheidet der Host-Header, welche Site antwortet. Ohne passenden Header landet die Prüfung bei der Standardwebsite oder bei gar keiner, und du bekommst ein 404 oder ein 400 „Bad Request – Invalid Hostname“ für einen Server, der eigentlich gesund ist.

Die Lösung heißt „Use HTTP/1.1“ zusammen mit einem eingetragenen Host, etwa portal.contoso.de. Ohne eigenen Eintrag verwendet der LoadMaster laut Dokumentation die IP-Adresse des Virtual Service, was dem IIS genauso wenig hilft. Alternativ lässt sich der Host-Header auch über die eigenen Header überschreiben. Bei HTTPS kommt eine zweite Ebene hinzu: Ist die Bindung mit „SNI erforderlich“ angelegt, braucht schon der TLS-Handshake den richtigen Namen. Prüfe bei deiner Firmware, ob und wie der Check den Hostnamen auch als SNI mitsendet. Wenn nicht, bleibt dir eine zusätzliche Bindung ohne SNI-Pflicht für den Check oder eine Probe auf einem eigenen Port.

TIPP • Erst von Hand prüfen, dann klicken

Bevor du am LoadMaster rätst, stell die Frage des Checks selbst, und zwar direkt an den Real Server vorbei am Virtual Service. Unter Windows hilft curl.exe, das seit Windows 10 mitgeliefert wird. Mit --resolve zwingst du den Namen auf die Adresse des einzelnen Servers, mit -I schickst du wie der Check eine HEAD-Anfrage:

```
curl.exe -k -I --resolve portal.contoso.de:443:10.0.20.11
https://portal.contoso.de/probe/health.aspx
curl.exe -k -I -H "Host: mail.contoso.de" https://10.0.20.21/owa/healthcheck.htm
```

Der erste Aufruf simuliert einen Check mit korrektem Namen inklusive SNI, der zweite einen Check mit Host-Header, aber ohne SNI. Liefern die beiden unterschiedliche Ergebnisse, hast du dein Problem gefunden, bevor du die erste Option im LoadMaster angefasst hast.

HEAD, GET und das Muster im Antworttext

HEAD ist die sparsamste Methode: Der Server liefert nur Statuszeile und Header. Für die meisten Probe-URLs genügt das, weil der Statuscode die Information trägt. Willst du zusätzlich den Inhalt prüfen, brauchst du GET und das Feld für das Antwortmuster, das nur bei einer 200-Antwort ausgewertet wird. Das ist nützlich, wenn eine Anwendung im Fehlerfall zwar 200 liefert, aber eine freundliche Fehlermeldung in den Text schreibt. Solche Anwendungen sind leider verbreiteter, als man hofft.

Mit einem vorangestellten Ausrufezeichen kehrst du das Muster um: Der Server gilt dann als krank, wenn der Text vorkommt. Daraus lässt sich ein eleganter Wartungsschalter bauen. Die Probe-Seite liefert normalerweise „OK“, und wer einen Server in die Wartung schicken will, ändert eine Zeile in einer Konfigurationsdatei, sodass sie „MAINTENANCE“ liefert. Der LoadMaster nimmt den Server aus der Verteilung, ohne dass jemand die WUI öffnen muss. Für das geordnete Ausleeren bestehender Sitzungen bleibt das Deaktivieren des Real Servers am LoadMaster trotzdem die sauberere Methode.

WICHTIG · Authentifizierte Checks nur mit Bedacht

Der LoadMaster kann über einen eigenen Authorization-Header auch angemeldet prüfen. Laut Dokumentation setzt das HTTP/1.1 und HEAD voraus, und der Wert ist ein Base64-kodiertes „Benutzer:Kennwort“ mit Verfahren davor, also Basic-Authentifizierung. Für Windows-Webdienste bedeutet das: Du müsstest Basic auf dem Server aktivieren und ein Kennwort in die LoadMaster-Konfiguration schreiben, das in jedem Backup landet. Das ist selten eine gute Idee. Eine anonyme Probe-Seite ist fast immer die bessere Wahl.

Intervalle, Timeouts, Retries: wie schnell ist tot?

Die drei Zeitparameter entscheiden, wie lange deine Benutzer mit einem kaputten Server verbracht haben, bevor der LoadMaster es merkt. Und umgekehrt, wie nervös dein LoadMaster auf eine Garbage Collection mit Gedenkminute reagiert.

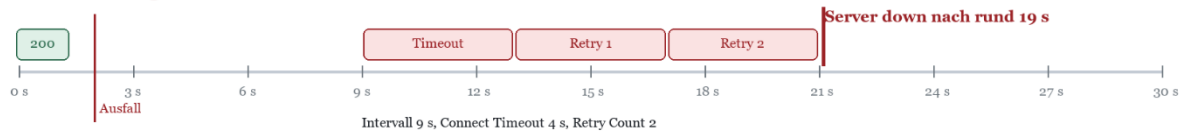
Parameter	Standard	Bereich	Bedeutung
Check Interval	9 s	9 bis 901 s	Abstand zwischen zwei Prüfrunden. Untergrenze ist Connect Timeout mal Retry Count plus 1.
Connect Timeout	4 s	4 bis 60 s	Wie lange der LoadMaster auf Verbindung und Antwort wartet.
Retry Count	2	2 bis 15	Wiederholungen, wenn der Timeout zuschlägt.

Der entscheidende Satz aus der Herstellerdokumentation ist unscheinbar: Wiederholt wird nur, wenn der Timeout erreicht wird. Kommt irgendeine HTTP-Antwort zurück, sei es 200, 404 oder 503, gibt es in dieser Prüfrunde keinen weiteren Versuch. Ein Server, der ehrlich „503“ sagt, fliegt also beim ersten Mal raus. Ein Server, der hängt und gar nichts sagt, darf sich mit den Standardwerten noch zweimal blamieren, bevor er aus der Verteilung genommen wird.

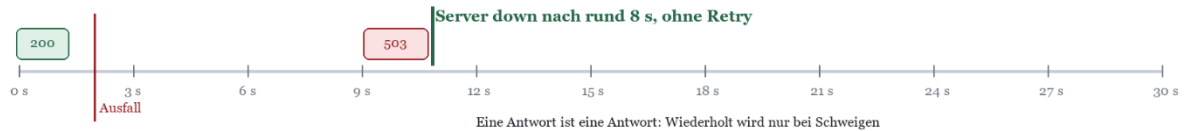
Wie schnell ist tot?

Timeout-Fehler werden wiederholt, eine falsche HTTP-Antwort nicht

Fall A: Server hängt, keine Antwort



Fall B: Server antwortet mit 503



Skizze: Erkennungszeit mit Standardwerten, vereinfacht; reale Zeiten hängen vom Takt ab

Für die Praxis heißt das: Ein sauber antwortender Health-Endpoint ist nicht nur aussagekräftiger, er ist auch schneller. Wer dagegen den Timeout hochdreht, weil die Probe-Seite manchmal fünf Sekunden braucht, verlängert die Erkennungszeit für den Fall „Server hängt“ gleich mit, denn das Intervall wächst automatisch mit. Bei 15 Sekunden Timeout und drei Retries bist du schnell bei einer Minute, in der ein toter Server noch Anfragen bekommt. Behebe lieber die langsame Probe-Seite.

FAKTEN • Global oder je Virtual Service

Bis Firmware 7.2.52 galten Intervall, Timeout und Retries für alle Virtual Services gemeinsam. Seitdem kannst du sie je Virtual Service oder SubVS überschreiben. Das ist praktisch, wenn ein träger Altdienst großzügigere Werte braucht, während Exchange und ADFS zügig reagieren sollen. Dokumentiere Abweichungen, sonst sucht in zwei Jahren jemand nach dem Grund, warum ausgerechnet ein Dienst so langsam umschaltet.

TIPP • Flattern vermeiden

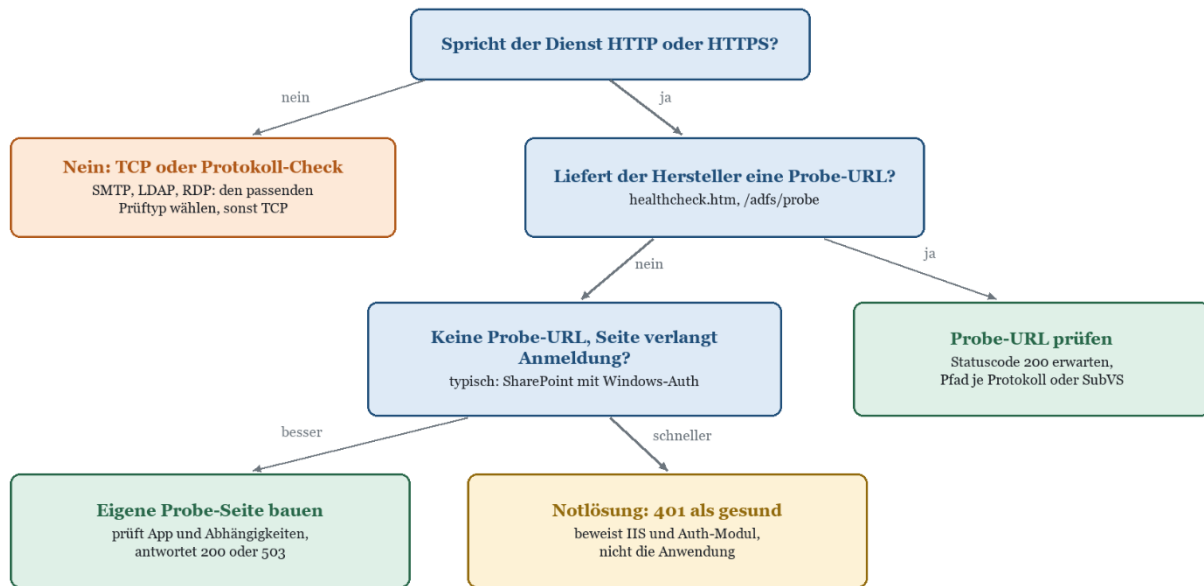
Ein Server, der im Sekundentakt zwischen „Up“ und „Down“ wechselt, ist schlimmer als ein Server, der einfach aus ist: Persistenz bricht, Sitzungen springen, Benutzer melden sich neu an. Wenn du Flattern siehst, ist fast nie der Check zu streng, sondern die Probe zu teuer oder der Server überlastet. Schau zuerst ins IIS-Log nach der Antwortzeit der Probe-URL, bevor du an den Zeitparametern drehst.

Praxis mit Microsoft-Brille: Exchange, SharePoint, ADFS

Jetzt wird es konkret. Die drei häufigsten Microsoft-Workloads hinter einem LoadMaster haben drei völlig unterschiedliche Vorstellungen davon, wie man ihren Gesundheitszustand abfragt. Exchange bringt eine durchdachte Lösung mit, ADFS eine minimalistische und SharePoint gar keine.

Welcher Check für welchen Dienst?

Faustregel: so tief prüfen, wie der Dienst es ehrlich zulässt



Skizze: Entscheidungsweg für die Wahl des Health Checks, vereinfacht

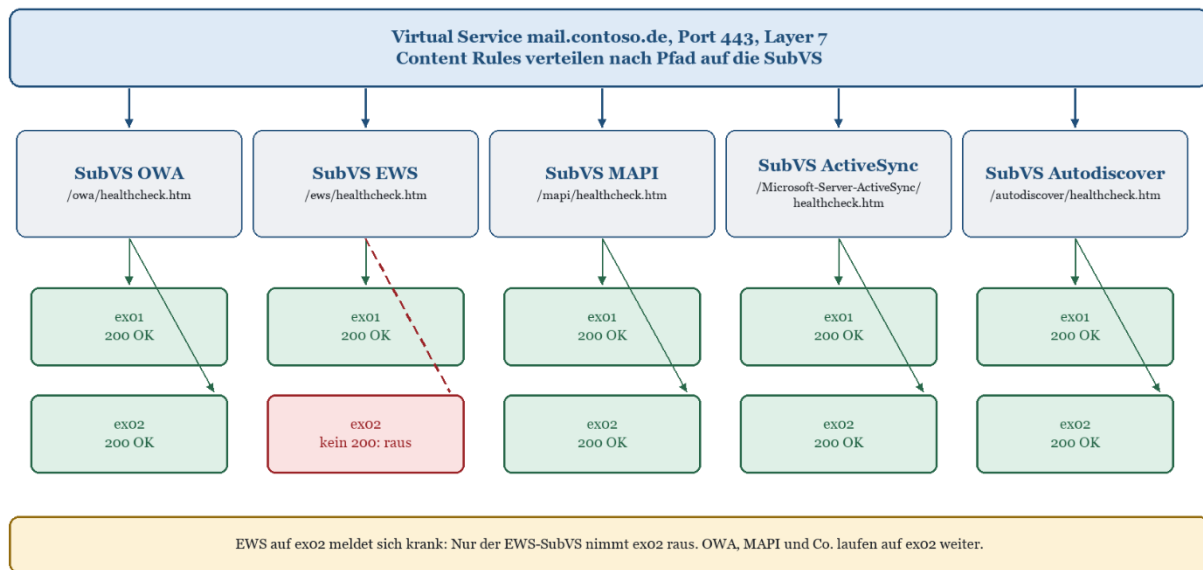
Exchange: healthcheck.htm je Protokoll

Seit Exchange 2013 überwacht Managed Availability die Server selbst und stellt das Ergebnis für Load Balancer bereit: Unter jedem virtuellen Verzeichnis gibt es eine Seite healthcheck.htm, die nicht als Datei auf der Platte liegt, sondern von Exchange erzeugt wird. Ist das Protokoll gesund, liefert sie 200 und eine kurze Zeile mit dem Servernamen. Hält Managed Availability das Protokoll für krank oder hast du die Komponente per Set-ServerComponentState inaktiv gesetzt, liefert sie keinen 200 mehr, und genau darauf soll der Load Balancer reagieren.

Der Clou ist die Granularität. OWA kann auf einem Server krank sein, während EWS und MAPI einwandfrei laufen. Ein einzelner Check für alle Protokolle wirft dann entweder einen Server ganz raus, der zu 80 Prozent funktioniert, oder er lässt ihn ganz drin, obwohl OWA dort tot ist. Deshalb gehört zu jedem Protokoll ein eigener SubVS mit eigenem Check auf das jeweilige healthcheck.htm. Wie SubVS und Content Rules zusammenspielen, erklärt der Beitrag [Virtual Service, Real Server, SubVS: das Grundvokabular des LoadMasters](#).

Exchange: jedes Protokoll fragt für sich

Ein Server, mehrere Gesundheitszustände. healthcheck.htm gibt es je virtuellem Verzeichnis



Skizze: Exchange-Prüfungen je Protokoll, Namen beispielhaft

Protokoll	Check-URL	Anmerkung
OWA	/owa/healthcheck.htm	Prüft den OWA-Pfad, nicht die Anmeldung
ECP	/ecp/healthcheck.htm	Oft gemeinsam mit OWA veröffentlicht, trotzdem eigener Check sinnvoll
EWS	/ews/healthcheck.htm	Wichtig für Outlook, Teams-Kalender, mobile Apps
MAPI über HTTP	/mapi/healthcheck.htm	Das Hauptprotokoll für Outlook
Outlook Anywhere	/rpc/healthcheck.htm	Nur, solange noch Clients es nutzen
ActiveSync	/Microsoft-Server-ActiveSync/healthcheck.htm	Mobilgeräte
OAB	/oab/healthcheck.htm	Offline-Adressbuch
Autodiscover	/autodiscover/healthcheck.htm	Fällt es aus, fällt die Profilkonfiguration aus

Den Statuscode stellst du auf „nur 200“ ein, also eine eigene Liste, die weder 301 und 302 noch 401 enthält. HTTP/1.1 mit dem Exchange-Namespace als Host-Header schadet nicht und erspart dir Ärger, falls auf den Servern weitere Bindungen hängen. Templates, Namespaces und das Zusammenspiel mit Wartungsfenstern vertieft der Beitrag [Exchange Server hinter dem LoadMaster: Templates, Namespaces, Health Checks](#).

TIPP · Wartungsmodus, den der LoadMaster versteht

Setzt du einen Exchange-Server mit Set-ServerComponentState für ServerWideOffline auf Inactive, melden seine healthcheck.htm-Seiten keinen 200 mehr, und der LoadMaster nimmt ihn von selbst aus allen SubVS. Das klappt aber nur, wenn deine Checks auch wirklich auf healthcheck.htm zeigen. Mit TCP-Checks bekommt ein Server im Wartungsmodus weiter Verkehr, und die Benutzer dürfen live dabei zusehen, wie du ihn neu startest.

SharePoint: 401 als „gesund“ werten oder eine eigene Probe-Seite

SharePoint Server bringt keine eigene Probe-URL für Load Balancer mit. Jede Seite einer Webanwendung mit Windows-Authentifizierung beantwortet eine anonyme Anfrage zunächst mit 401 und der Aufforderung zur Anmeldung. Weil 401 in den Standardcodes des LoadMasters enthalten ist, wird eine SharePoint-Farm mit Standardeinstellungen zwar grün, aber aus dem falschen Grund. Der 401 kommt nämlich vom Authentifizierungsmodul des IIS, bevor SharePoint selbst überhaupt gefragt wird. Ob die Inhaltsdatenbank erreichbar ist oder der App-Pool beim Laden hängt, erfährst du so nicht.

Variante	Was sie beweist	Aufwand	Bewertung
401 als gesund werten	IIS, Bindung und Auth-Modul leben	keiner	Notlösung für den Anfang
Statische Datei im Site-Verzeichnis	IIS liefert Dateien aus	gering	Kaum besser als 401, SharePoint bleibt unbeteiligt
Eigene Probe-Seite, anonym	App-Pool lädt, SharePoint antwortet, optional Datenbankzugriff	mittel	Empfehlung
Angemeldeter Check mit Basic	Volle Kette inklusive Anmeldung	hoch, Sicherheitsrisiko	Nicht empfohlen

Die Probe-Seite ist am Ende eine kleine Anwendungsseite, die anonym erreichbar ist, im Kontext von SharePoint läuft und je nach Ergebnis 200 oder 503 liefert. Ob du dafür eine anonyme Zone, eine eigene Erweiterung der Webanwendung oder einen separaten Pfad verwendest, hängt von deiner Farm und deinen Sicherheitsvorgaben ab. Wichtig ist nur: Die Seite darf nichts verraten außer „gesund“ oder „nicht gesund“, und sie sollte nicht von außen erreichbar sein. Eine Content Rule am Virtual Service, die den Pfad für Clients sperrt, erledigt das. Die Besonderheiten der Subscription Edition hinter dem LoadMaster beschreibt [SharePoint Server Subscription Edition hinter dem LoadMaster](#), das Hintergrundwissen zu SharePoint selbst findest du im [SharePoint-Wissensbereich](#).

AUS DER PRAXIS · Wenn 401 plötzlich krank ist

Ein typischer Fall: Eine Verwaltung stellt den SharePoint-Check auf „nur 200“, um es „richtig“ zu machen, lässt die URL aber auf „/“. Ergebnis: Alle Server sind sofort down, die Farm ist komplett weg, und der Kollege, der die Änderung gemacht hat, stellt um 7:45 Uhr auf TCP zurück. Der Check war nicht falsch, er war nur unvollständig. Wer den Statuscode verschärft, muss vorher die URL auf eine anonyme Probe-Seite umstellen. Reihenfolge ist alles, auch bei Health Checks.

ADFS und Web Application Proxy: /adfs/probe

ADFS macht es dir seit Windows Server 2012 R2 erfreulich einfach. Sowohl die ADFS-Server als auch der Web Application Proxy stellen den Endpunkt /adfs/probe bereit, und zwar über HTTP auf Port 80. Der Endpunkt antwortet mit 200, wenn der Dienst läuft und Anfragen annimmt. Gedacht ist er genau für Load Balancer, und er hat einen großen Vorteil: Er kommt ohne Anmeldung, ohne Zertifikat und ohne SNI aus. Damit umgeht er elegant die Zertifikatsbindungen von http.sys, an denen HTTPS-Checks auf ADFS gern scheitern.

Der Haken liegt genau in diesem Vorteil. Weil die Probe über Port 80 läuft, merkt sie nichts von einem abgelaufenen oder falsch gebundenen SSL-Zertifikat auf Port 443. Der LoadMaster meldet alles grün, während jeder Browser eine Zertifikatswarnung zeigt. Kombiniere die Probe deshalb mit einer Zertifikatsüberwachung in deinem Monitoring. Der Check im Virtual Service verwendet

dann einen abweichenden Prüfport, nämlich 80, während der Virtual Service selbst auf 443 lauscht. Mehr zur Farm, zur Probe und zu SNI-Fallstricken steht im Beitrag [ADFS und Web Application Proxy mit dem LoadMaster: Farm, Probe, SNI](#).

WARNUNG · Port 80 muss erreichbar sein, aber nur für den LoadMaster

Die ADFS-Probe setzt voraus, dass der LoadMaster die Server auf Port 80 erreicht. Gibt die Windows-Firewall den Port nicht frei oder blockiert eine Netzwerk-Firewall den Weg, ist jeder Server down, und der Reflex „dann eben TCP 443“ ist wieder nicht weit. Gib Port 80 auf den ADFS- und WAP-Servern gezielt für die Adressen des LoadMasters frei und veröffentliche ihn nicht nach außen.

RDS und Eigenentwicklungen in einem Satz

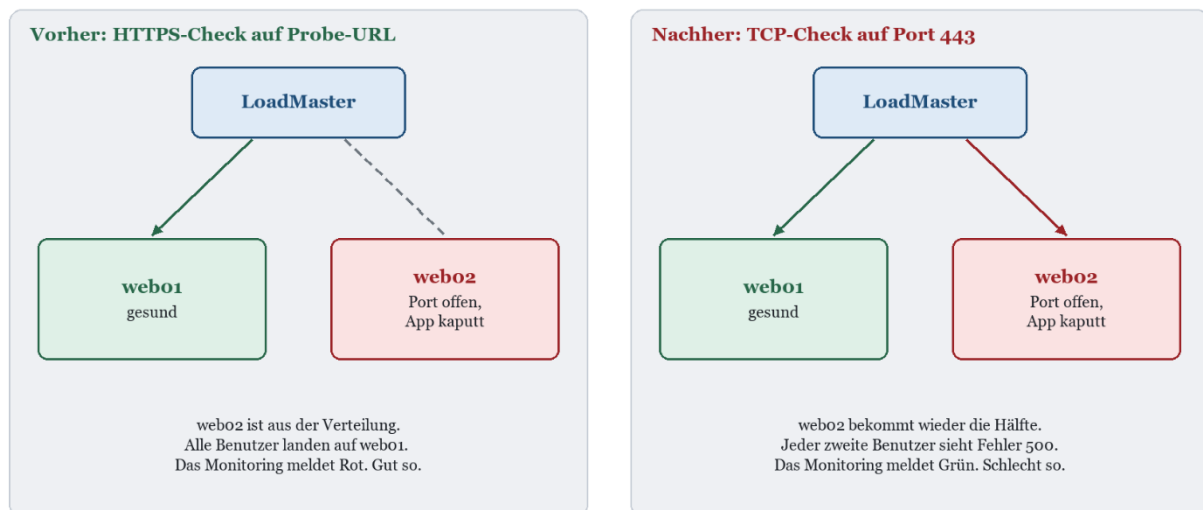
Bei RDP ohne HTTP bleibt dir oft nur TCP oder ein protokollnaher Check, und das ist dort auch ehrlich, solange du weißt, was er nicht prüft. RD Web und RD Gateway sprechen HTTPS und verdienen einen HTTP(S)-Check auf eine sinnvolle URL. Und wenn deine Entwicklung eine Webanwendung baut, bitte sie um einen Health-Endpoint, der Abhängigkeiten prüft und 200 oder 503 liefert. Das kostet eine Stunde Entwicklungszeit und erspart dir Jahre an Rätseln.

Der Klassiker: „Check rot? Dann eben TCP.“

Diese Geschichte spielt sich in so gut wie jedem Unternehmen irgendwann ab, meistens kurz vor Feierabend. Ein HTTPS-Check schlägt fehl, ein oder mehrere Server sind „Down“, Benutzer rufen an. Unter Druck stellt jemand den Check auf TCP um, und siehe da: Alles ist grün, die Anrufe hören auf. Also zumindest die Hälfte davon. Am nächsten Tag denkt niemand mehr daran, und der TCP-Check bleibt die nächsten drei Jahre in Produktion.

Der Klassiker: „Check rot? Stell halt auf TCP.“

Der Check war nicht kaputt. Er hatte recht.



Skizze: Rückfall auf TCP-Check bei halbtotem Server, Namen beispielhaft

Das Problem: In den meisten Fällen war der Check nicht kaputt, er hatte recht. Der Server war halbtot, der App-Pool lief nicht, ein Zertifikat war abgelaufen, eine Datenbank nicht erreichbar. Durch die Umstellung auf TCP bekommt dieser Server wieder einen fairen Anteil am Verkehr, und jeder Benutzer, der dort landet, sieht Fehler. Mit Persistenz sogar dauerhaft. Das Monitoring des LoadMasters zeigt Grün, das Monitoring der Benutzer zeigt Wut.

Erst diagnostizieren, dann umstellen

Wenn ein Check rot wird, gehe die Kette von unten nach oben durch, statt die Prüfung herunterzustufen:

- **Antwortet der Server überhaupt?** Ein curl.exe-Aufruf mit HEAD auf die Check-URL, direkt an den Server, zeigt dir den Statuscode, den der LoadMaster sieht.
- **Stimmt der Statuscode?** Ein 404 deutet auf Pfad oder Host-Header, ein 403 auf IP-Beschränkungen oder SSL-Einstellungen, ein 503 auf den App-Pool oder einen bewussten Wartungszustand.
- **Passt TLS?** Abgelaufene Zertifikate, fehlende SNI-Bindung oder abgeschaltete Protokollversionen nach einer Härtung brechen den HTTPS-Check, bevor HTTP überhaupt beginnt.
- **Ändert sich etwas mit dem Host-Header?** Wenn der Server mit Namen 200 liefert und mit IP 404, fehlt HTTP/1.1 mit Host im Check.
- **Hat sich am Server etwas geändert?** Updates, Härtung, neue Bindungen und Zertifikatswechsel sind die üblichen Verdächtigen. Das Windows-Ereignisprotokoll und das IIS-Log mit dem Zeitstempel des ersten Fehlschlags verraten meist mehr als jede LoadMaster-Anzeige.

Wie du am LoadMaster selbst mit Logs, L7-Debug und Paketmitschnitt nachsiehst, was der Check tatsächlich geschickt und bekommen hat, zeigt [Troubleshooting am LoadMaster: Logs, L7-Debug und TCP-Dump](#).

WICHTIG · Wenn es wirklich brennt

Brauchst du sofort eine Lösung, deaktiviere den kranken Real Server am LoadMaster, statt den Check zu entschärfen. Das Ergebnis für die Benutzer ist dasselbe, nur ehrlich: Der kaputte Server bekommt keinen Verkehr, und am nächsten Morgen sieht jeder sofort, dass da noch etwas offen ist. Ein entschärfter Check dagegen versteckt das Problem so gut, dass es erst beim nächsten Ausfall wieder auftaucht, dann gern auf dem zweiten Server.

Wenn du bei einer gewachsenen Konfiguration nicht mehr sicher bist, welcher Check was prüft und warum ein Dienst auf TCP steht, lohnt sich ein strukturierter Durchgang mit einem zweiten Paar Augen. Genau das ist ein typischer Einstieg bei der [Beratung zum Kemp LoadMaster](#).

FAQ zu Health Checks am Kemp LoadMaster

Reicht ein TCP-Check für Exchange nicht aus, wenn die Server sowieso überwacht werden?

Nein. Dein Monitoring alarmiert dich, aber es nimmt keinen Server aus der Verteilung. Solange der LoadMaster nur den Port prüft, schickt er weiter Benutzer auf einen kranken Server, bis jemand eingreift. Mit healthcheck.htm je Protokoll reagiert der LoadMaster selbst, innerhalb von Sekunden.

Warum wird mein Server als „Up“ angezeigt, obwohl die Anwendung Fehler liefert?

Weil die Standardliste 301, 302 und 401 als gesund wertet und eine Prüfung auf „/“ häufig genau so eine Antwort bekommt. Setze eine eigene Statuscodeliste und eine Check-URL, die von der Anwendung selbst beantwortet wird.

Kann ich den Check auf einen anderen Port als den Virtual Service legen?

Ja, ein abweichender Prüfport ist vorgesehen und bei ADFS mit /adfs/probe auf Port 80 der Normalfall. Bedenke, dass du dann nur prüfst, was auf diesem Port läuft. TLS-Probleme auf dem eigentlichen Port bleiben unsichtbar.

Wie verhindere ich, dass ein kurzer Hänger einen Server aus der Verteilung wirft?

Über Connect Timeout und Retry Count. Wiederholt wird nur bei Zeitüberschreitung, eine klare Fehlerantwort nimmt den Server sofort heraus. Wenn deine Probe-Seite regelmäßig in den Timeout läuft, ist die Seite das Problem, nicht der Check.

Darf ich für SharePoint einfach 401 als gesund werten?

Als Übergangslösung ja, das ist ohnehin der Standard. Dauerhaft solltest du eine anonyme Probe-Seite einrichten, die im Kontext von SharePoint läuft. Sonst merkt der LoadMaster nicht, wenn SharePoint hinter der Anmeldung nichts mehr liefert.

Welcher Statuscode bedeutet bei healthcheck.htm „krank“?

Für den Load Balancer zählt nur: 200 ist gesund, alles andere nicht. Stelle die Statuscodeliste so ein, dass Weiterleitungen und 401 nicht als gesund gelten. Dann ist es egal, mit welchem Code Exchange im Einzelfall antwortet.

Fazit: Frag, was deine Benutzer fragen

Ein Health Check ist ein Versprechen an deine Benutzer: „Wenn ich dich auf diesen Server schicke, funktioniert er.“ Ein TCP-Check auf Port 443 hält dieses Versprechen bei Windows-Webdiensten fast nie, weil http.sys auch dann noch den Hörer abnimmt, wenn dahinter niemand mehr wohnt. Ein HTTPS-Check auf „/“ mit Standard-Statuscodes ist kaum besser. Erst die Kombination aus aussagekräftiger Check-URL, bewusst gesetzter Statuscodeliste und passendem Host-Header macht aus dem Check eine echte Prüfung.

Exchange liefert mit healthcheck.htm je Protokoll die Vorlage, ADFS mit /adfs/probe eine schlanke Variante, und bei SharePoint und Eigenentwicklungen lohnt sich die kleine Investition in eine eigene Probe-Seite. Und wenn der Check das nächste Mal rot wird: Glaub ihm erst einmal. Er ist der einzige Kollege, der sich nie aus Bequemlichkeit für Grün entscheidet. Wer das Thema mit dem Team in Ruhe und am eigenen Labor durcharbeiten will, findet den Rahmen dafür in der [Kemp-LoadMaster-Schulung](#).

WEITERLESEN · Weiter im Thema

- › [Virtual Service, Real Server, SubVS: das Grundvokabular des LoadMasters](#)
- › [Exchange Server hinter dem LoadMaster: Templates, Namespaces, Health Checks](#)
- › [ADFS und Web Application Proxy mit dem LoadMaster: Farm, Probe, SNI](#)
- › [SharePoint Server Subscription Edition hinter dem LoadMaster](#)
- › [Troubleshooting am LoadMaster: Logs, L7-Debug und TCP-Dump](#)