Remote Desktop Services: RD Gateway, RD Web und Connection Broker balancieren



WISSEN

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

› **Kemp LoadMaster**

BERATUNG

RDS-Farm hinter den LoadMaster bringen, UDP endlich zum Laufen bekommen oder den Broker hochverfügbar machen? Wir schauen gemeinsam drauf.

› **Beratung zum Kemp LoadMaster**

SCHULUNG

Virtual Services, Persistenz und Health Checks für Microsoft-Workloads im Labor durchspielen, bevor es das Homeoffice der ganzen Belegschaft tut.

› **Kemp-LoadMaster-Schulung**

Remote Desktop Services sind der Dienst, über den sich niemand beschwert, solange er läuft, und über den sich alle beschweren, sobald er ruckelt. Und ruckeln tut er gern. Nicht weil die Sitzungshosts zu klein wären, sondern weil irgendwo zwischen Homeoffice und Rechenzentrum ein Load Balancer steht, der nur die Hälfte des Protokolls kennt. Das Ergebnis kennst du: Die Verbindung steht, aber jeder Mausklick fühlt sich an wie ein Ferngespräch nach Neuseeland in den Achtzigern. Die Kolleginnen und Kollegen in der Buchhaltung nennen das „die RDS-Gedenksekunde“. Du nennst es UDP-Transport, der nicht ankommt.

Dabei ist eine RDS-Farm hinter dem LoadMaster kein Hexenwerk. Sie verlangt nur, dass man drei Rollen auseinanderhält, die in Diagrammen gern im selben Kasten landen: das RD Gateway, das den Verkehr aus dem Internet annimmt und tunnelt, RD Web, das die Anmeldeseite und die Verbindungsdateien liefert, und den Connection Broker, der entscheidet, wer auf welchem Sitzungshost landet. Zwei davon gehören hinter den LoadMaster. Einer nicht. Und genau an dieser Stelle wird in Projekten am kreativsten falsch gebaut.

Dieser Beitrag gehört zur Serie rund um den Kemp LoadMaster und schließt die Lücke zwischen dem RDS-Deployment-Guide von Progress Kemp und den RDS-Seiten auf Microsoft Learn. Es geht um das RD Gateway mit HTTPS und UDP 3391, um die Persistenz, die beide Kanäle zusammenhält, um RD Web als ganz normale Webanwendung mit ein paar Eigenheiten, um den Connection Broker, der kein Load Balancing braucht, sondern DNS und SQL, und um die Verteilung auf die Sitzungshosts, die du lieber dem Broker überlässt. Am Ende steht der Praxisfall,

mit dem fast jedes Projekt irgendwann zu tun hat: UDP fehlt, alles läuft über HTTP, und alle sind langsam unglücklich.

FAKTEN · Das Wichtigste in sechs Sätzen

Das RD Gateway nutzt TCP 443 für den HTTPS-Transport und optional UDP 3391 für den UDP-Transport, der seit RDP 8 Bildaufbau und Eingaben deutlich flüssiger macht.

Beide Kanäle einer Sitzung müssen auf demselben Gateway landen; ohne gekoppelte Persistenz verwirft der falsche Gateway den UDP-Verkehr, und der Client fällt still auf HTTPS zurück.

Auch der HTTP-Transport selbst besteht aus zwei Kanälen, einem eingehenden und einem ausgehenden; ein Load Balancer ohne Affinität zerreißt die Verbindung.

RD Web ist eine IIS-Anwendung mit Formularanmeldung; Persistenz ist empfehlenswert, und alle Server müssen dasselbe Zertifikat und dieselbe Konfiguration tragen.

Der Connection Broker wird seit Windows Server 2012 aktiv/aktiv betrieben, mit einer gemeinsamen SQL-Datenbank und einem DNS-Namen, der per Round Robin auf alle Broker zeigt.

Die Verteilung auf die Sitzungshosts einer Sammlung übernimmt der Broker; ein Virtual Service davor ist in einer Broker-Umgebung überflüssig und meistens schädlich.

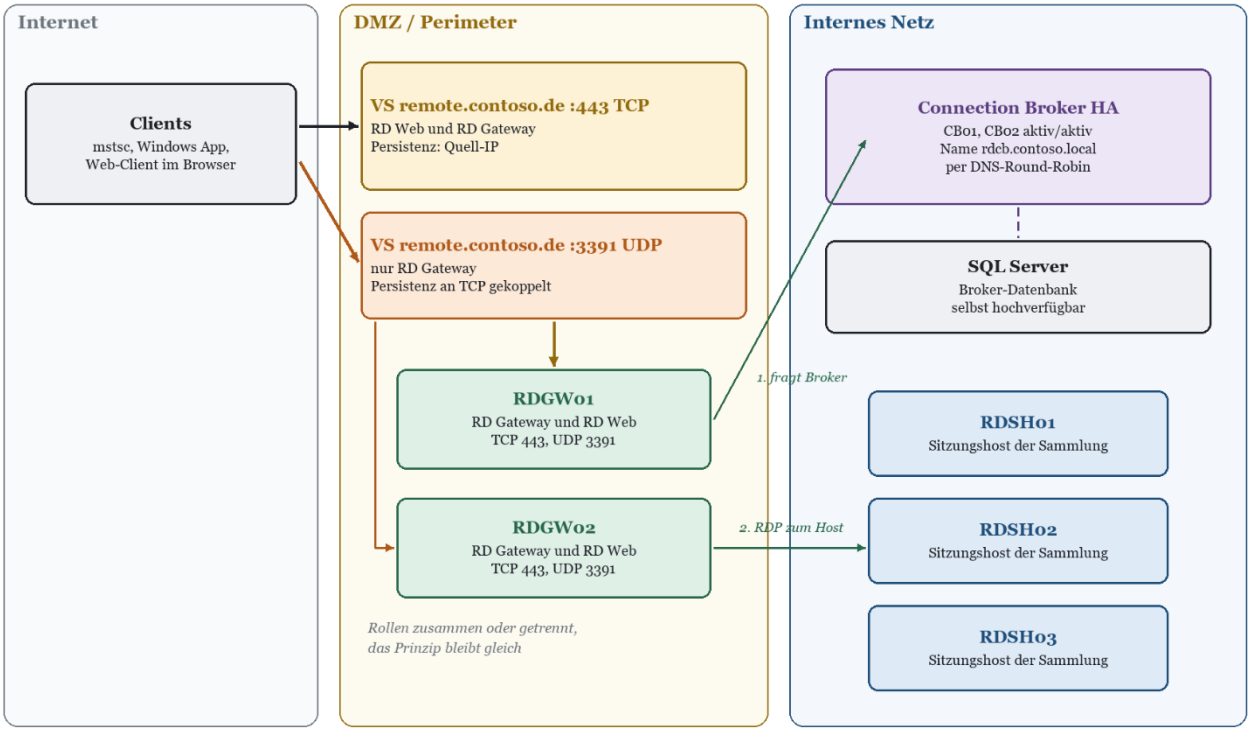
Die Architektur: drei Rollen, zwei Virtual Services, ein öffentlicher Name

Eine typische RDS-Umgebung im Mittelstand oder in der Verwaltung sieht so aus: Zwei Server mit RD Gateway und RD Web stehen im Perimeter oder im internen Netz hinter einer Firewall, zwei Connection Broker teilen sich eine SQL-Datenbank, und dahinter arbeiten einige Sitzungshosts in einer oder mehreren Sammlungen. Von außen gibt es genau einen Namen, zum Beispiel remote.contoso.de. Unter diesem Namen holt sich der Browser die RD-Web-Seite, und unter demselben Namen baut der Client später den Gateway-Tunnel auf.

Für den LoadMaster bedeutet das: eine VIP, zwei Virtual Services. Einer nimmt TCP 443 an und bedient sowohl RD Web als auch das Gateway, weil beide auf denselben Servern und demselben Port liegen. Der zweite nimmt UDP 3391 an und kennt nur das Gateway. Wer die Rollen auf getrennte Server verteilt, kann RD Web und Gateway auch über unterschiedliche Namen oder per SubVS trennen; am Prinzip ändert das nichts. Was ein Virtual Service, ein Real Server und ein SubVS genau sind, erklärt der Beitrag Virtual Service, Real Server, SubVS: das Grundvokabular des LoadMasters.

Die Referenzarchitektur: wer wird balanciert, wer balanciert selbst

Ein öffentlicher Name remote.contoso.de, dahinter zwei Virtual Services an derselben VIP



Skizze: Referenzaufbau, Namen fiktiv; HA-Paar der LoadMaster nicht dargestellt

Rolle	Protokoll und Port	Gehört hinter den LoadMaster?	Persistenz
RD Gateway	TCP 443 (HTTPS), UDP 3391	Ja, beide Ports	Pflicht, Quell-IP, TCP und UDP gekoppelt
RD Web Access	TCP 443 (HTTPS)	Ja	Empfohlen, Quell-IP oder Cookie
RD-Web-Client (HTML5)	TCP 443 über das Gateway	Ja, über Gateway und RD Web	wie Gateway
Connection Broker	TCP 3389, intern	Nein, HA über SQL und DNS	entfällt
Sitzungshosts	TCP und UDP 3389, intern	Nein, der Broker verteilt	entfällt
Lizenzserver	RPC, intern	Nein, zwei Server in der Konfiguration	entfällt

WICHTIG · Ein Name, ein Zertifikat, viele Stellen

Der öffentliche Name taucht in einer RDS-Bereitstellung an mehreren Stellen auf: im Zertifikat der Gateway- und RD-Web-Server, in den Bereitstellungseigenschaften als Gateway-Servername und in jeder .rdp-Datei, die RD Web ausliefert. Trägst du dort einen einzelnen Servernamen statt des Namens der VIP ein, umgeht jeder Client den LoadMaster und landet zuverlässig immer auf demselben Gateway. Das fällt erst auf, wenn genau dieser Server in Wartung geht.

TLS: durchreichen, re-verschlüsseln oder bridgen

Beim Gateway hast du mehr Freiheit als etwa bei ADFS, aber nicht beliebig viel. Am einfachsten ist ein Virtual Service, der TLS unverändert durchreicht. Dann hält jeder Gateway sein eigenes Zertifikat, und der LoadMaster sieht nur TCP und Quell-IP. Willst du am LoadMaster in den Verkehr schauen, etwa für Content Rules oder eine Vorauthentifizierung von RD Web, terminierst du TLS und verschlüsselst zum Gateway neu. Das RD Gateway kennt dafür in seinen Eigenschaften das sogenannte SSL-Bridging, mit dem es Verbindungen von einem vorgelagerten Gerät per HTTPS oder auch per HTTP annimmt. Reines HTTP zum Gateway spart Rechenzeit, verlagert aber die Verantwortung für die Verschlüsselung im internen Netz komplett auf dich. In Verwaltungen mit BSI-Grundschutz-Brille ist das meistens schnell beantwortet.

Der UDP-Kanal ist von alledem nicht betroffen: Er ist per DTLS verschlüsselt, und der LoadMaster reicht ihn auf Layer 4 durch. Ein UDP-Offload gibt es hier nicht, und du willst auch keinen. Die Grundlagen zu Offload, Re-Encryption und Zertifikatsketten stehen im Beitrag TLS-Offload, Re-Encryption und Zertifikate: PKI trifft LoadMaster.

RD Gateway: HTTPS, UDP 3391 und die Persistenz, die beides zusammenhält

Das RD Gateway ist im Kern ein Tunnel. Der Client verpackt RDP in HTTPS, schickt es an Port 443, und das Gateway packt es wieder aus und reicht es an den Sitzungshost weiter. Seit Windows Server 2012 und RDP 8 kann das Gateway zusätzlich einen UDP-Kanal auf Port 3391 anbieten. Der Client baut zuerst die HTTPS-Verbindung auf, authentifiziert sich, erhält die Autorisierung für den Sitzungshost und versucht danach, einen UDP-Seitenkanal zum selben Gateway aufzubauen. Klappt das, läuft der Großteil des Bild- und Eingabeverkehrs über UDP, und HTTPS bleibt als Steuerkanal bestehen. Klappt es nicht, läuft eben alles über HTTPS. Ohne Fehlermeldung, ohne Warnung, nur langsamer.

Warum der HTTP-Transport schon allein Persistenz braucht

Selbst wenn du UDP komplett ignorierst, kommt das Gateway nicht ohne Affinität aus. Der HTTP-Transport des RD Gateway besteht aus zwei HTTP-Verbindungen, einem eingehenden und einem ausgehenden Kanal, die zusammen eine Sitzung bilden. Beide müssen auf demselben Gateway landen, denn nur dieser Server kennt den Kontext. Verteilt der LoadMaster die zweite Verbindung per Round Robin auf den anderen Server, bricht der Aufbau ab, und der Client meldet einen Fehler, der wahlweise nach Zertifikat, nach Netzwerk oder nach Mondphase klingt. Genau deshalb hat Microsoft DNS-Round-Robin für Gateway-Farmen mit dem HTTP-Transport nie empfohlen: DNS kennt keine Affinität.

Die bewährte Lösung am LoadMaster ist Persistenz nach Quell-IP-Adresse. Sie ist unspektakulär, funktioniert mit durchgereichtem TLS und lässt sich auf den UDP-Service übertragen. Cookie-basierte Persistenz klingt eleganter, setzt aber TLS-Terminierung voraus und hilft beim UDP-Kanal überhaupt nicht, denn UDP-Pakete haben keine Cookies. Welche Persistenzarten der LoadMaster kennt und wann welche sinnvoll ist, beschreibt der Beitrag Persistence und Scheduling: wer hängt an welchem Server und warum.

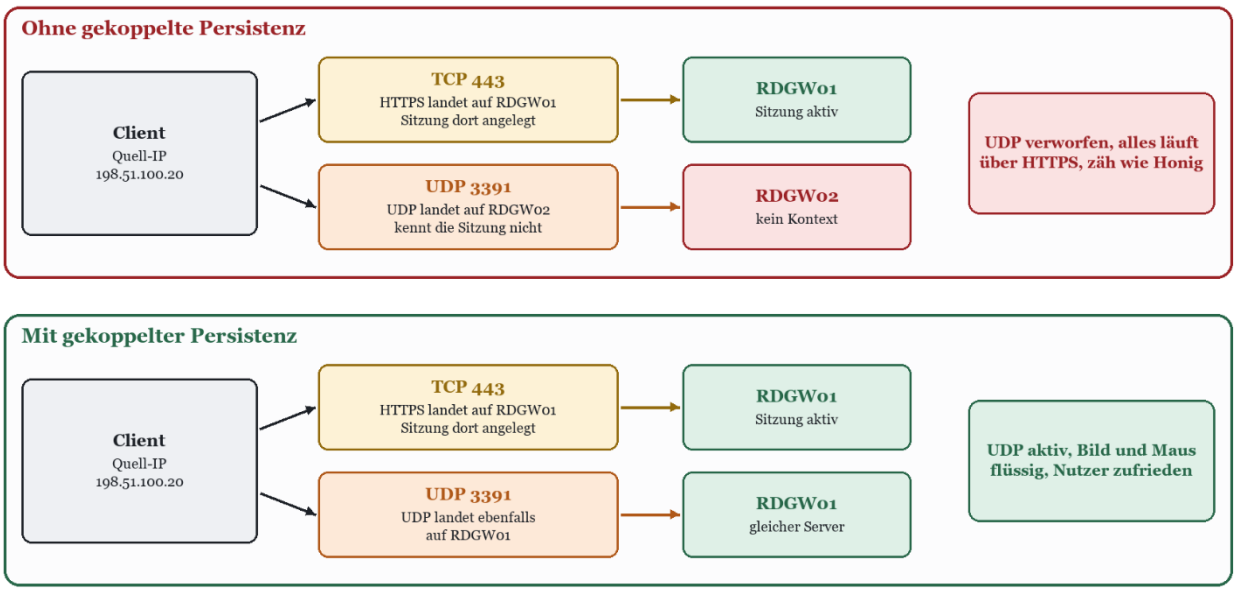
UDP 3391: der zweite Virtual Service und seine Kopplung

Für den UDP-Kanal legst du einen eigenen Virtual Service an: gleiche VIP, Protokoll UDP, Port 3391, dieselben Real Server wie beim TCP-Service. Der Deployment Guide von Progress Kemp sieht auch hier Persistenz nach Quell-IP vor. Entscheidend ist aber nicht, dass beide Services Persistenz haben, sondern dass sie für denselben Client dieselbe Entscheidung treffen. Zwei unabhängige Persistenztabellen können, gerade nach einem Failover oder nach Ablauf eines Timeouts, unterschiedliche Real Server wählen. Der LoadMaster bietet dafür eine Funktion zur Kopplung der Persistenz über Virtual Services hinweg, die in der WUI als Port Following geführt

wird. Prüfe in deiner Firmware-Version, unter welchen Bedingungen sie für die Kombination aus TCP und UDP greift; die Voraussetzungen zu Persistenzmodus und VIP haben sich über die Versionen hinweg verändert.

Warum UDP ohne gekoppelte Persistenz ins Leere läuft

Der UDP-Kanal gehört zu einer Sitzung, die nur der Gateway kennt, der die HTTPS-Verbindung hält



Skizze: Prinzip; Adressen aus dem Dokumentationsbereich

Einstellung	VS TCP 443	VS UDP 3391
Protokoll	TCP, TLS durchgereicht oder re-verschlüsselt	UDP, Layer 4
Real Server	alle Gateway-Server, Port 443	dieselben Gateway-Server, Port 3391
Persistenz	Quell-IP	Quell-IP, an TCP-Service gekoppelt
Persistenz-Timeout	länger als eine typische Sitzung	identisch zum TCP-Service
Scheduling	Least Connection	Least Connection
Health Check	HTTPS, mehr als nur Port offen	keiner sinnvoll möglich, Status vom TCP-Service ableiten
Weiterleitung	SNAT oder transparent	SNAT oder Direct Server Return, siehe Praxisteil

WARNUNG · Der Persistenz-Timeout ist keine Kosmetik

RDS-Sitzungen dauern Stunden. Ist der Persistenz-Timeout am LoadMaster kürzer als eine Sitzung mit Mittagspause, vergisst der LoadMaster den Client, während der Tunnel noch steht. Baut der Client dann einen Kanal neu auf, etwa nach einem kurzen WLAN-Aussetzer, landet er womöglich auf dem anderen Gateway. Das HTTPS-Paar bricht, der UDP-Kanal verschwindet, und die Wiederverbindung dauert. Zu lang ist auch nicht gut: Ein Gateway, das nach der Wartung zurückkommt, bekommt dann stundenlang keine Benutzer. Eine Stunde, wie im Kemp-Guide, ist ein vernünftiger Startwert; justiere nach deinen echten Sitzungsmustern.

Health Check für das Gateway

Ein offener Port 443 sagt beim Gateway erstaunlich wenig. IIS antwortet auch dann fröhlich, wenn der Dienst TSGateway steht, das Zertifikat abgelaufen ist oder der Netzwerkrichtlinienserver für die Autorisierung nicht erreichbar ist. Ein HTTPS-Check, der eine echte Antwort des Gateway-Pfads erwartet, ist besser als ein reiner Verbindungstest; die vollständige Kette aus Anmeldung, Autorisierung und Tunnel prüft aber nur eine synthetische Anmeldung aus dem Monitoring. Für den UDP-Service gibt es keinen sinnvollen Inhaltscheck, deshalb sollte sein Real-Server-Status dem des TCP-Service folgen. Wie man Prüfungen baut, die mehr können als „Port offen“, steht in Health Checks, die wirklich prüfen: warum „Port 443 offen“ nicht reicht.

TIPP · Wartung ohne Herzrasen

Setze einen Gateway vor Updates im LoadMaster auf deaktiviert, und zwar in beiden Virtual Services. Bestehende Sitzungen laufen weiter, neue Verbindungen gehen auf den Partner. Warte, bis die Zahl der Verbindungen im RD Gateway Manager gegen null geht, und starte erst dann neu. Wer nur den TCP-Service deaktiviert, schickt neue UDP-Kanäle weiter auf den Server, der gleich neu startet.

RD Web: die Webanwendung, die alle für trivial halten

RD Web Access ist eine ASP.NET-Anwendung unter /RDWeb im IIS. Sie zeigt die Anmeldeseite, listet die veröffentlichten Desktops und RemoteApps und liefert für jeden Klick eine .rdp-Datei mit Gateway-Name, Sammlung und Einstellungen aus. Seit Windows Server 2016 kommt der RD-Web-Client dazu, der RDP im Browser per HTML5 abbildet und dafür zwingend über das Gateway geht. Für den LoadMaster ist RD Web eine normale HTTPS-Anwendung. Normal heißt aber nicht anspruchslos.

Persistenz und Machine Key

Die klassische RD-Web-Seite arbeitet mit Formularanmeldung und einem Cookie, das ASP.NET mit dem Machine Key der jeweiligen Anwendung schützt. Landet ein angemeldeter Benutzer beim nächsten Klick auf dem anderen Server, kann dieser das Cookie nur lesen, wenn beide Server denselben Machine Key nutzen. Sonst steht der Benutzer wieder vor der Anmeldeseite und hält das für eine Sicherheitsfunktion. Die saubere Lösung ist, den Machine Key für /RDWeb und für den Feed-Pfad auf allen RD-Web-Servern identisch zu setzen. Die pragmatische Lösung ist Persistenz am LoadMaster. Die robuste Lösung ist beides. Weil RD Web und Gateway im Mittelstand meist auf denselben Servern liegen, erbt RD Web die Quell-IP-Persistenz des Gateway-Service ohnehin.

Health Check für RD Web

Der Pfad /RDWeb leitet weiter, und eine Weiterleitung ist kein Gesundheitsnachweis. Prüfe stattdessen die Anmeldeseite selbst, die unter /RDWeb/Pages/ mit einem Sprachunterordner und login.aspx liegt, und erwarte 200 OK. Damit weißt du immerhin, dass IIS, der Anwendungspool und ASP.NET leben. Ob die Ressourcenliste nach der Anmeldung befüllt ist, verrät dir das nicht: Kann RD Web den Broker nicht erreichen, ist die Seite grün und der Benutzer sieht trotzdem eine leere Liste.

Vorauthentifizierung und MFA

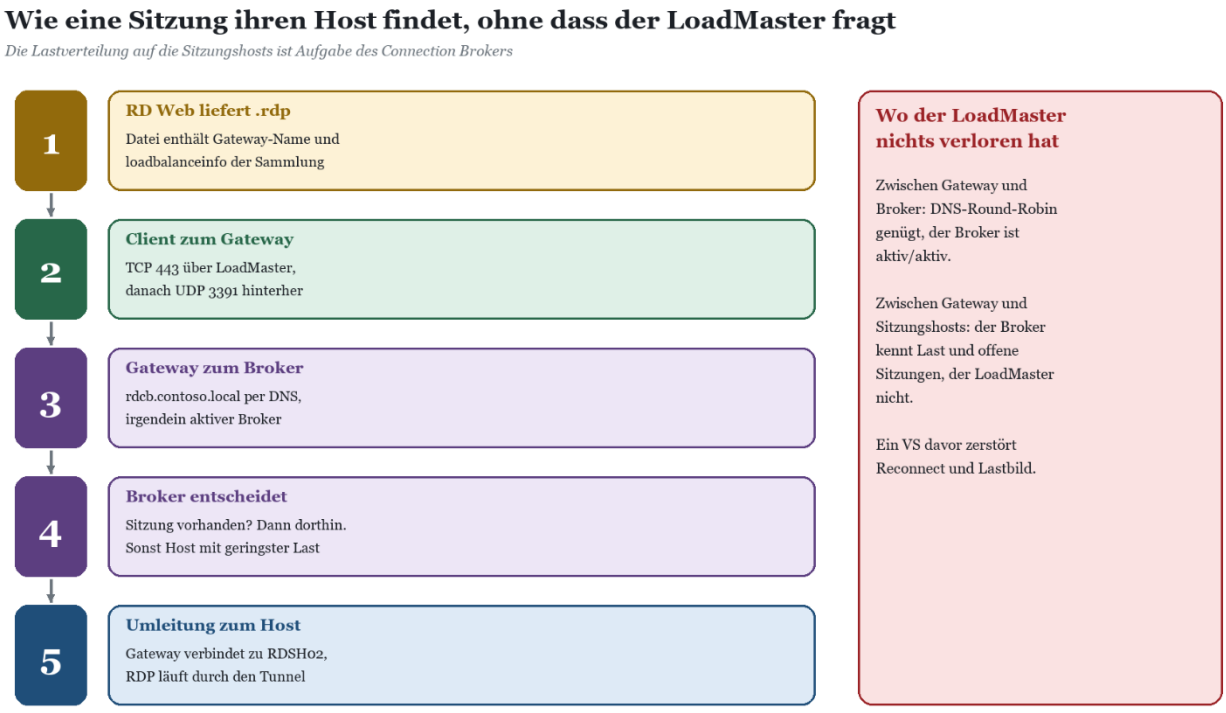
RD Web ist die offensichtliche Stelle für eine zusätzliche Anmeldung, etwa per Edge Security Pack oder SAML. Die Ernüchterung folgt beim Gateway: Der eigentliche Tunnel wird von mstsc oder der Windows App aufgebaut, nicht vom Browser, und dieser Client kann mit einer Formularanmeldung am LoadMaster nichts anfangen. Eine Vorauthentifizierung vor RD Web schützt also die Seite, nicht den Zugang. MFA für den Tunnel selbst gehört an das Gateway, klassisch über den Netzwerkrichtlinienserver mit einer MFA-Erweiterung. Was am LoadMaster geht und was nicht, steht in MFA vor OWA, ECP und Co.: Zwei-Faktor am LoadMaster.

AUS DER PRAXIS · Die Anmeldeseite, die sich immer wieder zeigt

Eine Verwaltung hatte RD Web auf zwei Servern, Least Connection, keine Persistenz, weil „die Webseite ja zustandslos ist“. Die Benutzer meldeten sich an, klickten auf ihren Desktop und landeten in etwa der Hälfte der Fälle wieder auf der Anmeldeseite. Die Fehlersuche dauerte zwei Tage, weil der Effekt beim Test mit einem einzelnen Browser ausgerechnet nie auftrat. Die Ursache: unterschiedliche Machine Keys. Die Lösung: zehn Minuten im IIS-Manager und Quell-IP-Persistenz als Gürtel zu den Hosenträgern.

Connection Broker und Sitzungshosts: wo der LoadMaster die Finger weglässt

Jetzt kommt der Teil, der in Projekten die meisten Diskussionen auslöst. Der Connection Broker ist das Gehirn der Bereitstellung. Er kennt alle Sitzungshosts, ihre Last, die offenen und getrennten Sitzungen jedes Benutzers und die Zuordnung zu Sammlungen. Wenn ein Client eine Verbindung anfordert, entscheidet der Broker: Hat der Benutzer schon eine getrennte Sitzung, geht es genau dorthin. Wenn nicht, wählt der Broker den Host mit der geringsten Last, gewichtet nach der Konfiguration der Sammlung. Diese Entscheidung kann ein Load Balancer davor nicht besser treffen, weil er die Sitzungen schlicht nicht kennt.



Skizze: vereinfachter Ablauf einer Verbindung in eine Sitzungssammlung

Broker-Hochverfügbarkeit: SQL und DNS statt Virtual Service

Seit Windows Server 2012 lässt sich der Broker aktiv/aktiv betreiben. Alle Broker schreiben in eine gemeinsame SQL-Datenbank, wahlweise auf SQL Server oder, seit Windows Server 2016, auch auf Azure SQL. Für die Clients und für das Gateway bekommt die Broker-Farm einen gemeinsamen Namen, etwa rdcb.contoso.local, der per DNS-Round-Robin auf alle Broker zeigt. Eingerichtet wird das im Server-Manager oder per PowerShell mit Set-RDConnectionBrokerHighAvailability. Fällt ein Broker aus, versucht der Client oder das Gateway einfach die nächste Adresse. Mehr Hochverfügbarkeit braucht es an dieser Stelle nicht, und ein LoadMaster dazwischen bringt vor

allem eine weitere Fehlerquelle, ein weiteres Zertifikatsthema und eine weitere Stelle, an der Kerberos für den Broker-Namen funktionieren muss.

Die eigentliche Schwachstelle der Broker-HA ist nicht der Broker, sondern die Datenbank. Ein einzelner SQL Server ohne Always-On-Verfügbarkeitsgruppe oder Cluster macht aus zwei Brokern einen. Wer hier spart, hat eine sehr redundante Vorderseite und einen sehr einsamen Keller.

FAKTEN · Was der Broker wirklich braucht

Eine gemeinsame Datenbank auf hochverfügbarem SQL Server oder Azure SQL, den SQL Native Client beziehungsweise ODBC-Treiber auf jedem Broker, ein Computerkonto pro Broker mit Rechten auf die Datenbank, einen DNS-Namen mit einem A-Eintrag pro Broker und ein Zertifikat in den Bereitstellungseigenschaften, das diesen Namen enthält. Einen Load Balancer braucht er nicht.

Sitzungshosts verteilen: Aufgabe des Brokers, nicht des LoadMasters

Der Deployment Guide von Progress Kemp beschreibt auch Virtual Services für TCP 3389 vor den Sitzungshosts, samt einer Persistenzart, die das RDP-Cookie beziehungsweise den Routing-Token des Brokers auswertet. Das hat seine Berechtigung, aber in einem engen Feld: bei Sitzungshosts ohne Broker, bei älteren Umgebungen, die noch auf einen Broker im Umleitungsmodus setzen, oder bei Spezialfällen, in denen Clients direkt per IP auf eine Farmadresse zugreifen sollen. In einer modernen Bereitstellung mit Sammlungen und RD Web verteilt der Broker, und die .rdp-Datei enthält über den Parameter loadbalanceinfo die Information, zu welcher Sammlung der Client will. Ein Virtual Service vor den Hosts sieht dann nur noch Verbindungen, die das Gateway bereits an einen konkreten Host geschickt hat.

Szenario	Wer verteilt?	Rolle des LoadMasters
Sammlung mit Broker, Zugriff über Gateway	Broker	nur Gateway und RD Web
Sammlung mit Broker, interner Zugriff ohne Gateway	Broker	keine; Clients sprechen den Broker-Namen an
Sitzungshosts ohne Broker, gleiche Images	LoadMaster	VS auf TCP 3389, Persistenz nach RDP-Cookie oder Quell-IP
Altumgebung mit Broker im Umleitungsmodus	Broker plus LoadMaster	VS auf 3389 mit Routing-Token-Auswertung
Azure Virtual Desktop	AVD-Dienst	keine; RDS-Logik liegt beim Dienst

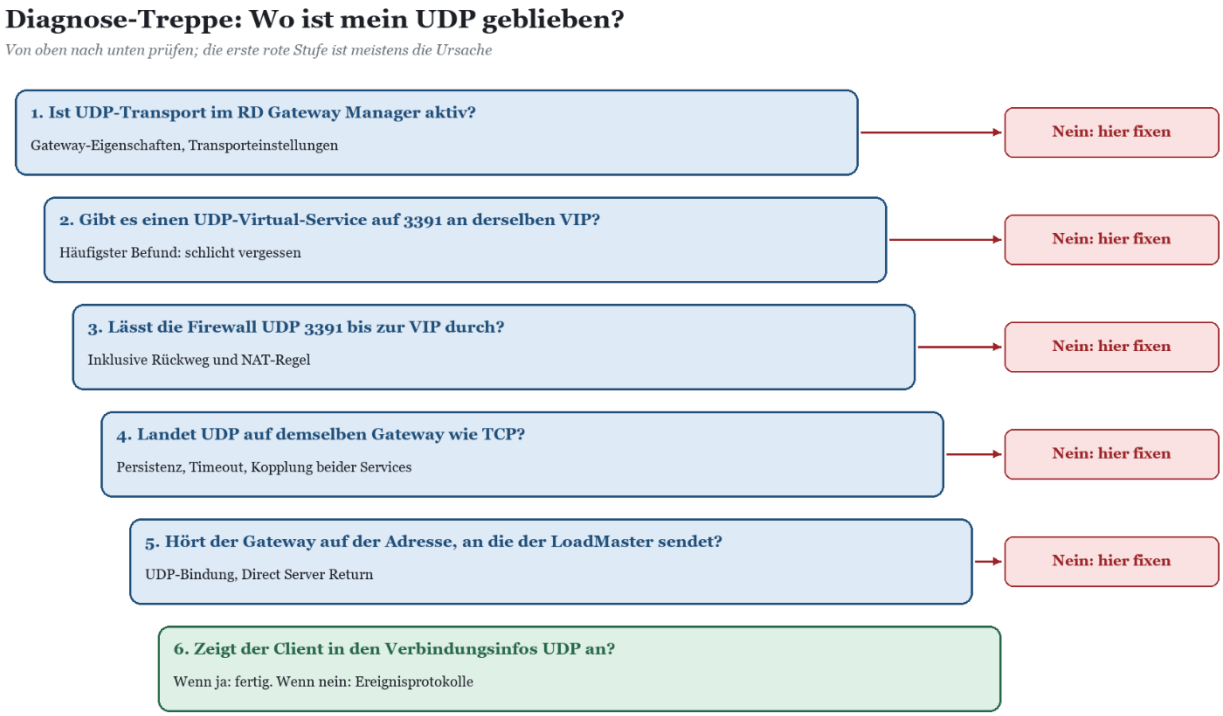
WARNUNG · Der Virtual Service vor dem Broker

Ein Klassiker aus der Kategorie „gut gemeint“: Jemand legt einen Virtual Service vor die beiden Broker, weil die Firewall-Regel so schöner aussieht. Danach funktionieren Wiederverbindungen nur noch manchmal, die Ereignisprotokolle füllen sich mit Umleitungsfehlern, und mit SNAT sieht der Broker alle Clients mit der Adresse des LoadMasters. Wenn ein Broker ausfällt, merkt der LoadMaster es oft später als DNS-Round-Robin es ohnehin ausgeglichen hätte. Rückbau ist die Lösung.

Praxisfall: UDP fehlt, alles läuft zäh über HTTP

Das Muster ist immer gleich. Die Umstellung auf den LoadMaster ist durch, die Verbindungen kommen zustande, niemand meldet einen Fehler. Nach einer Woche kommen die ersten Beschwerden: Videos ruckeln, Scrollen in Excel zieht Schlieren, die CAD-Abteilung droht mit

Rückkehr ins Büro. Klickt man in mstsc auf das Signalsymbol in der Verbindungsleiste, steht da, dass UDP nicht aktiviert ist. Willkommen im HTTPS-only-Betrieb. Der ist sicher, stabil und auf Strecken mit Latenz oder Paketverlust spürbar langsamer, weil TCP jeden verlorenen Bildaktualisierungsschnipsel brav nachliefert, auch wenn ihn längst niemand mehr braucht.



Skizze: Prüfreihenfolge aus der Praxis, keine Herstellervorgabe

Die üblichen Verdächtigen

- **Kein UDP-Virtual-Service.** Der häufigste Befund. Das Template wurde für TCP 443 angelegt, der UDP-Service nie. Der Client versucht UDP, bekommt keine Antwort und gibt nach kurzer Zeit auf.
- **Firewall blockt 3391.** Die Portfreigabe für 443 war selbstverständlich, für UDP 3391 hat niemand einen Antrag gestellt. Auch vorgelagerte NAT-Regeln und Provider-Router gehören auf die Liste.
- **Persistenz nicht gekoppelt.** TCP landet auf dem einen, UDP auf dem anderen Gateway. Der zweite Gateway kennt die Sitzung nicht und verwirft den Verkehr. Das Fiese daran: Es funktioniert zufällig in der Hälfte der Fälle, was jede Fehlersuche wunderbar verwirrt.
- **UDP-Bindung am Gateway.** Im RD Gateway Manager lässt sich festlegen, auf welchen Adressen der UDP-Transport lauscht. Bei Direct Server Return schickt der LoadMaster die Pakete an die VIP-Adresse, die auf dem Gateway als Loopback konfiguriert ist; lauscht der UDP-Transport nur auf der primären Adresse, ignoriert er sie. Die Einstellung auf alle nicht zugewiesenen Adressen löst das.
- **Asymmetrischer Rückweg.** Bei transparenter Weiterleitung muss die Antwort des Gateways über den LoadMaster zurück. Kennt der Gateway eine andere Route ins Internet, geht die UDP-Antwort an der VIP vorbei, und der Client verwirft sie, weil sie von der falschen Adresse kommt.
- **Unterwegs blockiert.** Hotel-WLAN, Gastnetze und manche Unternehmens-Proxys lassen kein UDP durch. Dagegen hilft kein LoadMaster der Welt; hier ist der HTTPS-Rückfall genau das, wofür er gebaut wurde.

Gerade der asymmetrische Rückweg ist kein RDS-Spezifikum, sondern die Folge der Entscheidung zwischen SNAT und Transparenz. Die Zusammenhänge beschreibt Client-IP sichtbar machen: Transparenz, SNAT und X-Forwarded-For.

So weist du es nach

Auf der Clientseite zeigt die Verbindungsinformation in mstsc, ob UDP aktiv ist. Auf dem Gateway listet der RD Gateway Manager unter der Überwachung die Verbindungen samt Transportart, und das Ereignisprotokoll unter Microsoft-Windows-TerminalServices-Gateway verrät, ob ein UDP-Kanal aufgebaut oder abgelehnt wurde. Am LoadMaster zeigt die Statistik des UDP-Virtual-Service, ob überhaupt Pakete ankommen; bekommt er null, liegt das Problem vor dem LoadMaster. Bekommt er Pakete, aber der Gateway meldet nichts, hilft ein TCP-Dump auf der Schnittstelle zum Gateway, gefiltert auf Port 3391. Der vergleicht Quell-IP, Ziel-Real-Server und den Real Server, auf dem die zugehörige HTTPS-Verbindung läuft. Wie du Dumps und Logs am LoadMaster sauber ziehst, steht in Troubleshooting am LoadMaster: Logs, L7-Debug und TCP-Dump.

Symptom	Wahrscheinliche Ursache	Abhilfe
UDP nie aktiv, VS-Statistik zeigt null	Firewall oder kein UDP-VS	Regel und VS anlegen
UDP manchmal aktiv, manchmal nicht	Persistenz nicht gekoppelt	Kopplung und identische Timeouts
UDP-VS sieht Pakete, Gateway nicht	UDP-Bindung oder Real-Server-Port falsch	Bindung auf alle Adressen, Port 3391 prüfen
UDP bricht nach Minuten weg	Idle-Timeout an Firewall oder LoadMaster	UDP-Timeouts angleichen
Nur extern ohne UDP	Gastnetze, Hotel, Mobilfunk-NAT	nichts; Rückfall auf HTTPS ist gewollt

TIPP · Den Erfolg messbar machen

Bevor du umbaust, notiere bei drei Testbenutzern die Round-Trip-Time und den Transport aus der Verbindungsinformation, gern auch über die Leistungsindikatoren RemoteFX Network auf dem Sitzungshost. Nach dem Umbau dasselbe noch einmal. Ein Vorher-nachher mit Zahlen beendet Diskussionen deutlich schneller als „fühlt sich besser an“.

FAQ: Häufige Fragen zu Kemp und RDS

Brauche ich für das RD Gateway zwingend einen UDP-Virtual-Service?

Nein, die Verbindungen funktionieren auch nur über HTTPS. Ohne UDP verzichtest du aber auf den schnelleren Transport, und auf Strecken mit Latenz oder Paketverlust merken die Benutzer das deutlich.

Welche Persistenz nehme ich für das RD Gateway am LoadMaster?

Quell-IP-Persistenz für TCP 443 und UDP 3391, mit identischem Timeout und gekoppelt, damit beide Kanäle eines Clients auf demselben Gateway landen. Eine Stunde ist ein vernünftiger Startwert.

Funktioniert Quell-IP-Persistenz, wenn viele Benutzer hinter einem NAT sitzen?

Ja, sie landen dann alle auf demselben Gateway. Das verteilt ungleichmäßig, bricht aber nichts. Bei großen Außenstellen hinter einer Adresse lohnt ein Blick auf die Lastverteilung.

Muss ich den Connection Broker hinter den LoadMaster stellen?

Nein. Der Broker wird seit Windows Server 2012 aktiv/aktiv mit gemeinsamer SQL-Datenbank betrieben und per DNS-Round-Robin angesprochen. Ein Virtual Service davor erzeugt eher Probleme bei Umleitung und Wiederverbindung.

Soll der LoadMaster die Sitzungshosts einer Sammlung verteilen?

In einer Bereitstellung mit Broker nicht. Der Broker kennt Last und getrennte Sitzungen und entscheidet besser. Ein Virtual Service auf 3389 ist nur in Umgebungen ohne Broker oder in Sonderfällen sinnvoll.

Darf ich TLS am LoadMaster vor dem RD Gateway terminieren?

Ja, mit Re-Encryption zum Gateway oder mit SSL-Bridging in den Gateway-Eigenschaften. Einfacher und für die meisten Umgebungen ausreichend ist es, TLS durchzureichen.

Warum lande ich in RD Web nach der Anmeldung wieder auf der Anmeldeseite?

Weil der nächste Aufruf auf einem anderen RD-Web-Server landet, der das Anmelde-Cookie wegen eines abweichenden Machine Keys nicht lesen kann. Setze den Machine Key auf allen Servern identisch und aktiviere Persistenz.

Fazit: zwei Rollen balancieren, eine in Ruhe lassen

Eine RDS-Farm hinter dem LoadMaster folgt drei einfachen Regeln. Erstens: Das RD Gateway bekommt zwei Virtual Services, TCP 443 und UDP 3391, mit gekoppelter Quell-IP-Persistenz und einem Timeout, der zu echten Arbeitstagen passt. Zweitens: RD Web ist eine Webanwendung mit Anmelde-Cookie, also gleicher Machine Key, Persistenz und ein Health Check auf die Anmeldeseite statt auf die Weiterleitung. Drittens: Connection Broker und Sitzungshosts verteilen sich selbst, über SQL, DNS und die Logik des Brokers, und der LoadMaster schaut dabei freundlich zu.

Wenn du dich an diese Aufteilung hältst, verschwindet die RDS-Gedenksekunde in den meisten Umgebungen von allein. Und wenn nicht, weißt du jetzt, wo du suchen musst: fast immer beim UDP-Kanal, der irgendwo zwischen Firewall, Persistenz und Bindung verloren gegangen ist. Die gute Nachricht dabei: Er ist selten tot. Er hat sich nur verlaufen.

TIPP · Unterstützung bei Planung und Umsetzung

Wenn du deine RDS-Farm hinter den LoadMaster bringen, den UDP-Transport zum Laufen bekommen oder Broker und Gateway sauber hochverfügbar machen willst, hilft dir die Beratung zum Kemp LoadMaster. Wer Virtual Services, Persistenz und Health Checks mit dem eigenen Team im Labor durchspielen möchte, findet das in der Kemp-LoadMaster-Schulung.

WEITERLESEN · Weiter im Thema

› Persistence und Scheduling: wer hängt an welchem Server und warum
› Health Checks, die wirklich prüfen: warum „Port 443 offen“ nicht reicht
› TLS-Offload, Re-Encryption und Zertifikate: PKI trifft LoadMaster
› HA-Paar, Failover und Konfig-Sync: zwei LoadMaster, ein Gedanke
› Troubleshooting am LoadMaster: Logs, L7-Debug und TCP-Dump