



Guide to Deploying Microsoft Exchange 2013 with Citrix NetScaler

Extensive guide covering details of NetScaler ADC deployment with Microsoft Exchange 2013.

Table of Contents

Introduction	3
Exchange Server 2013 roles	3
Load balancing in Exchange 2013	4
Health Monitoring	5
Namespace and affinity scenarios	5
Single namespace / Layer 4 (no session affinity)	6
Single namespace / Layer 7 (no session affinity)	6
Single namespace / session affinity	7
Multiple namespaces / no session affinity	7
Product versions and prerequisites	8
Deploying Exchange 2013 with NetScaler	9
Solution features	9
Exchange 2013 deployment and configuration	10
Exchange 2013 deployment topology	10
Exchange 2013 configuration	10
Conclusion	17

This guide focuses on deploying Microsoft Exchange 2013 with Citrix NetScaler. NetScaler is a world-class application delivery controller (ADC) with the proven ability to load balance, accelerate, optimize and secure enterprise applications. Exchange, one of the most critical enterprise applications, provides access to email —the lifeline of any business.

Exchange 2013 brings a rich set of technologies, features and services to the Exchange Server product line. The goal is to support people and organizations as their work habits evolve from communication focused to collaboration focused. At the same time, Exchange 2013 helps lower total cost of ownership, whether you deploy it on premises or provision your mailboxes in the cloud. New features and functionality in Exchange 2013 are designed to do the following:

- **Support a multigenerational workforce.** Social integration and ease of finding people are important to users. Smart Search learns from user communication and collaboration behavior to enhance and prioritize search results in Exchange. Also, with Exchange 2013, users can merge contacts from multiple sources to provide a single view of a person by linking contact information pulled from multiple locations.
- **Provide an engaging experience.** Microsoft Outlook 2013 and Microsoft Office Outlook Web App have a fresh, new look. Outlook Web App emphasizes a streamlined user interface that also supports the use of touch, enhancing the mobile device experience with Exchange.
- **Integrate with SharePoint and Lync.** Exchange 2013 offers greater integration with Microsoft SharePoint 2013 and Microsoft Lync 2013 through site mailboxes and In-Place eDiscovery.
- **Help meet evolving compliance needs.** Compliance and eDiscovery are challenging for many organizations. Exchange 2013 helps you to find and search data not only in Exchange, but across your organization. With improved search and indexing, you can search across Exchange 2013, Lync 2013, SharePoint 2013 and Windows file servers.
- **Provide a resilient solution.** Exchange 2013 builds upon the Exchange Server 2010 architecture and has been redesigned for simplicity of scale, hardware utilization and failure isolation.

Exchange Server 2013 roles

The multi-role server architecture introduced with Exchange Server 2007, and continued with Exchange 2010, has been consolidated in Exchange Server 2013.

Exchange 2013 has three server roles that can be installed:

- Client Access server
- Mailbox server
- Edge Transport server (from SP1 or later)

Load balancing in Exchange 2013

Load balancing has been at the core of any Exchange deployment from the beginning. The major change with Exchange 2013 is that it no longer requires session affinity to be maintained at the load balancer. To understand this better and see how it impacts your Exchange 2013 design and deployment, here is the sample protocol flow:

1. Client resolves the namespace to a virtual IP address hosted on the load balancer.
2. The load balancer assigns the session to a CAS member in the load balanced pool.
3. CAS authenticates the request and does service discovery to retrieve
 1. Mailbox version
 2. Mailbox location information
4. CAS makes a decision on whether to proxy the request or redirect the request to another CAS infrastructure.
5. CAS queries an Active Manager instance that is responsible for the database to determine which mailbox server is hosting the active copy.
6. CAS proxies the request to the Mailbox server hosting the active copy.

Step 5 is the fundamental change that removes the need for session affinity at the load balancer. For a given protocol session, CAS now maintains a 1:1 relationship with the Mailbox server that is hosting user data. In the event that the active database copy is moved to a different Mailbox server, CAS closes the sessions to the previous server and establishes sessions to the new server. This means that all sessions, regardless of their origination point (i.e., CAS members in the load balanced array), end up at the same place, the Mailbox server hosting the active database copy.

The protocol used in step 6 depends on the protocol used to connect to CAS. If the client leverages the HTTP protocol, then the protocol used between the CAS and Mailbox server is HTTP (secured via SSL using a self-signed certificate). If the protocol leveraged by the client is IMAP or POP, then the protocol used between the CAS and Mailbox server is IMAP or POP.

Telephony requests are unique, however. Instead of proxying the request at step 6, CAS will redirect the request to the Mailbox server hosting the active copy of the user's database, as the telephony devices support redirection and need to establish their SIP and RTP sessions directly with the unified messaging components on the Mailbox server.

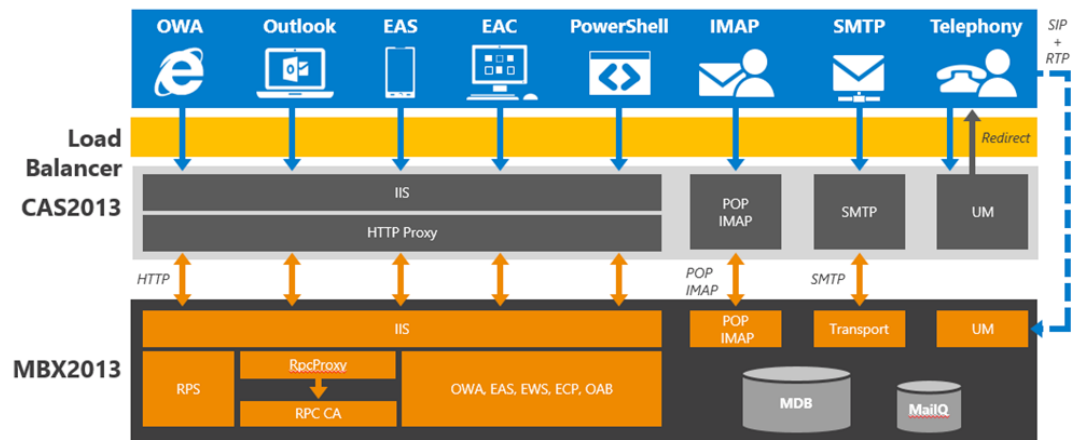


Figure 1. Exchange 2013 Client Access protocol architecture

Health Monitoring

Exchange 2013 includes a built-in monitoring solution known as Managed Availability. Managed Availability includes an offline responder. When the offline responder is invoked, the affected protocol (or server) is removed from service. To ensure that load balancers do not route traffic to a CAS that Managed Availability has marked as offline, load balancer health probes must be configured to check <virtualdirectory>/healthcheck.htm (e.g., <https://mail.contoso.com/owa/healthcheck.htm>.) Note that healthcheck.htm does not actually exist within the virtual directories; it is generated in memory based on the component state of the protocol in question.

If the load balancer health probe receives a 200 status response, then the protocol/server is up; if the load balancer receives a different status code, then Managed Availability has marked that protocol instance down on the CAS. As a result, the load balancer should also consider that endpoint down and remove the CAS from the applicable load balancing pool.

Namespace and affinity scenarios

Now that we understand how health checks are performed, let's look at four scenarios:

1. Single namespace / Layer 4 (no session affinity)
2. Single namespace / Layer 7 (no session affinity)
3. Single namespace / session affinity
4. Multiple namespaces / no session affinity

Single namespace / Layer 4 (no session affinity)

In this scenario, a single namespace is deployed for all HTTP protocol clients (mail.contoso.com). The load balancer is operating at Layer 4 and is not maintaining session affinity. The load balancer is also configured to check the health of the target CAS in the load balancing pool; however, because this is a Layer 4 solution, the load balancer is configured to check the health of only a single virtual directory

(as it cannot distinguish OWA requests from RPC requests). Administrators will have to choose which virtual directory they want to target for the health probe; they should choose a virtual directory that is heavily used. For example, if the majority of your users utilize OWA, then targeting the OWA virtual directory in the health probe is appropriate.

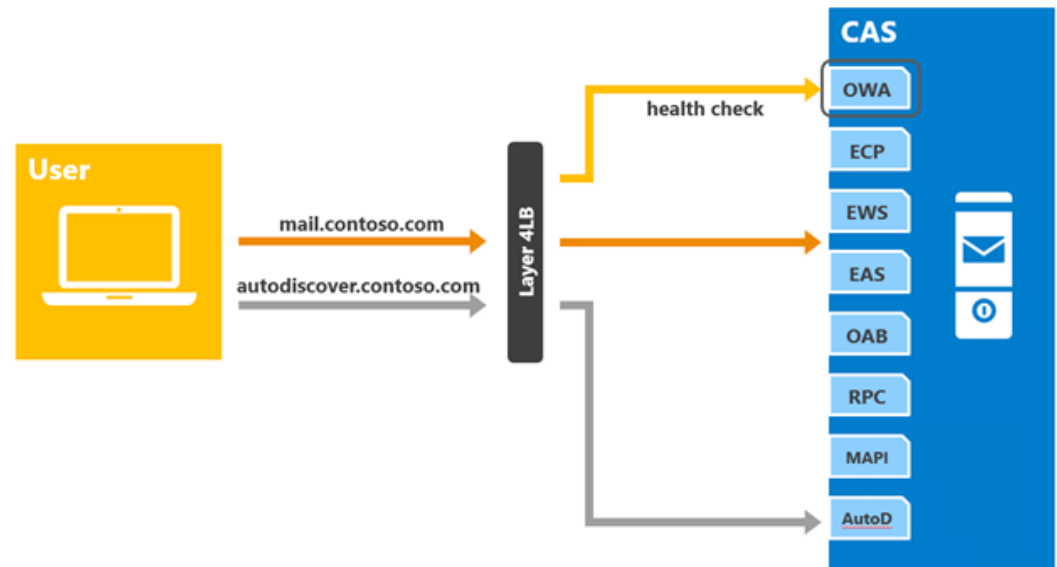


Figure 2. Single namespace with no session affinity

As long as the OWA health probe response is healthy, the load balancer will keep the target CAS in the load balancing pool. However, if the OWA health probe fails for any reason, then the load balancer will remove the target CAS from the load balancing pool for all requests associated with that particular namespace. In other words, in this example, health from the perspective of the load balancer is per-server, not per-protocol, for the given namespace. This means that if the health probe fails, all client requests must be directed to another server, regardless of protocol.

Single namespace / Layer 7 (no session affinity)

In this scenario, a single namespace is deployed for all the HTTP protocol clients (mail.contoso.com). The load balancer is configured to utilize Layer 7, meaning SSL termination occurs and the load balancer knows the target URL. The load balancer is also configured to check the health of the target CAS in the load balancing pool; in this case, a health probe is configured on each virtual directory.

As long as the OWA health probe response is healthy, the load balancer will keep the target CAS in the OWA load balancing pool. However, if the OWA health probe fails for any reason, the load balancer will remove the target CAS from the load balancing pool for OWA requests. In other words, in this example, health is per protocol; this means that if the health probe fails, only the affected client protocol will have to be directed to another server.

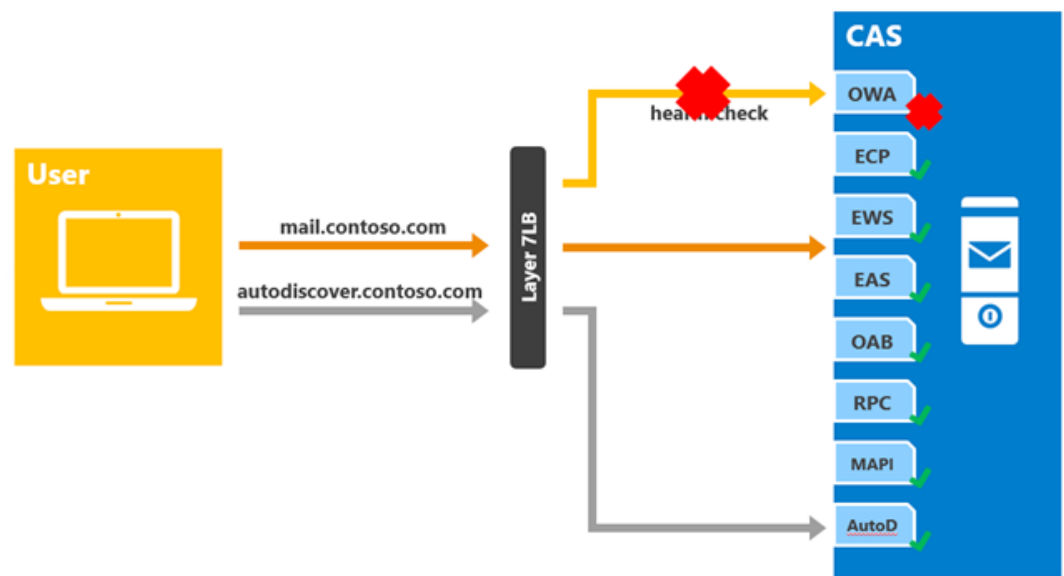


Figure 3. Single namespace with Layer 7 (no session affinity) - health probe failure

Single namespace / session affinity

In this scenario, a single namespace is deployed for all HTTP protocol clients (`mail.contoso.com`). The load balancer is configured to maintain session affinity (Layer 7), meaning SSL termination occurs and the load balancer knows the target URL. The load balancer is also configured to check the health of the target CAS in the load balancing pool; in this case, the health probe is configured on each virtual directory.

As long as the OWA health probe response is healthy, the load balancer will keep the target CAS in the OWA load balancing pool. However, if the OWA health probe fails for any reason, the load balancer will remove the target CAS from the load balancing pool for OWA requests. In other words, in this example, health is per protocol; this means that if the health probe fails, only the affected client protocol will have to be directed to another server.

Multiple namespaces / no session affinity

This scenario combines the best of both worlds – it provides a per-protocol health check while not requiring complex load balancing logic.

In this scenario, a unique namespace is deployed for each HTTP protocol client; for example:

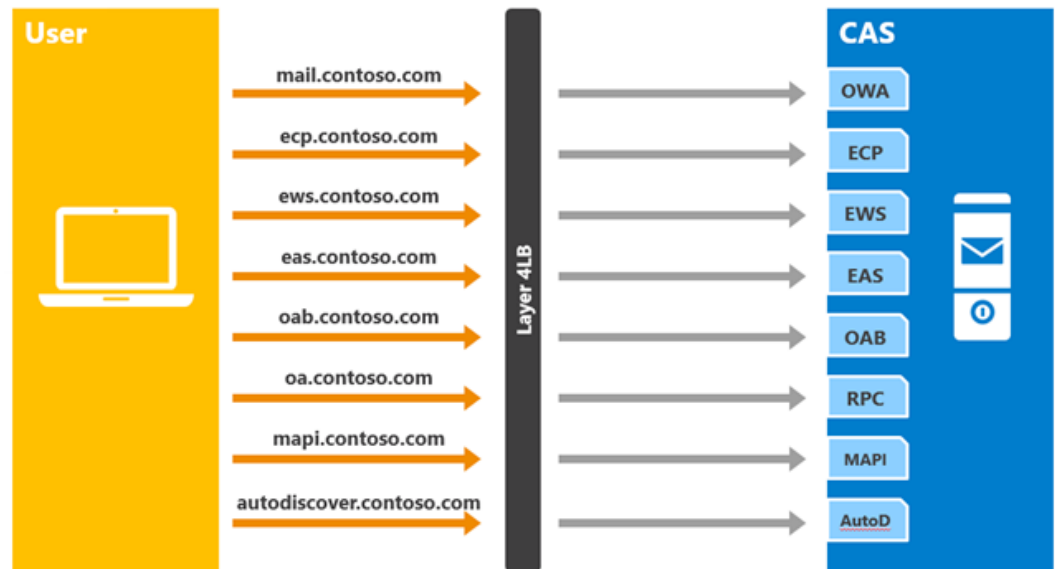


Figure 4. Multiple namespaces with no session affinity

The load balancer is configured so session affinity (Layer 4) is not maintained. The load balancer is also configured to check the health of the target CAS in the load balancing pool. In this case, the health probes are configured to target the health of each virtual directory, as each virtual directory is defined with a unique namespace.

As long as the OWA health probe response is healthy, the load balancer will keep the target CAS in the OWA load balancing pool. However, if the OWA health probe fails for any reason, the load balancer will remove the target CAS from the load balancing pool for OWA requests. In other words, in this example, health is per protocol; this means that if the health probe fails, only the affected client protocol will have to be directed to another server.

The downside to this approach is that it introduces additional namespaces and additional VIPs (one per namespace), and increases the number of names added as subject alternative names on the certificate, which can be expensive depending on your certificate provider. But this approach does not introduce extra complexity to the end user – the only URL the user needs to know is the OWA URL. ActiveSync, Outlook and Exchange Web Services clients will utilize Autodiscover to determine the correct URL.

Product versions and prerequisites

Product	Version
Microsoft Exchange	Exchange 2013
License	Enterprise Edition
NetScaler® ADC	Release 9.3 and above
License	Enterprise

Deploying Exchange 2013 with NetScaler

Solution features

The following NetScaler features are used in Exchange 2013 deployment. Please ensure these features are enabled in the NetScaler system.

- Content switching
- Load balancing
- Health monitoring
- SSL offload

Here is a quick explanation of how these features work.

Content switching

The content switching module directs incoming traffic to an optimal matching load balancing virtual server. This logical switching of incoming traffic based on content type allows you to configure specific optimization policies.

Load balancing

NetScaler load balancing evenly distributes requests to backend servers. Multiple algorithms are supported to provide efficient load balancing logic for every application server.

Health monitoring

NetScaler load health monitoring ensures that only backend servers in good state are selected after the load balancing decision is made. Intelligent monitoring of backend servers prevents requests from being sent to malfunctioning application servers.

SSL offload

SSL connections are terminated at the NetScaler appliance. This process allows NetScaler to conduct advanced traffic monitoring discussed in this deployment guide. Additionally, SSL offload can significantly reduce the computational overhead of offloading encrypted user connections on backend servers.

Exchange 2013 deployment and configuration

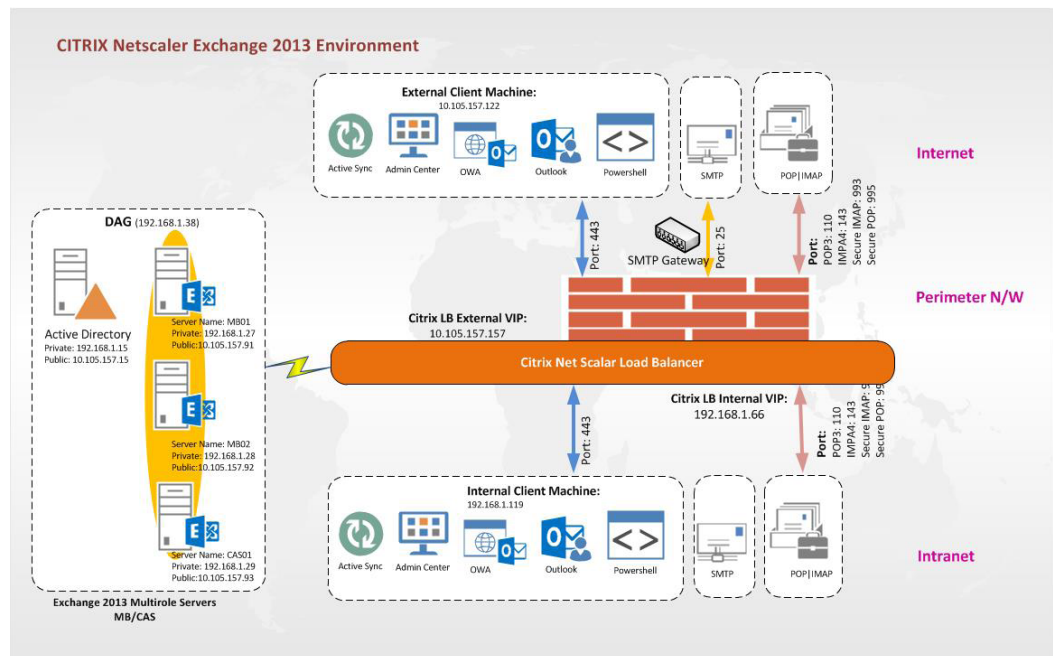


Figure 5. Exchange 2013 deployment topology

Exchange 2013 configuration

Note: For this configuration to work as described (with SSL enabled), you should enable SSL offloading for Exchange 2013. To enable this feature, please read the instructions provided at [https://technet.microsoft.com/en-us/library/dn635115\(v=exchg.150\).aspx](https://technet.microsoft.com/en-us/library/dn635115(v=exchg.150).aspx)

Service group configuration

Add service groups to manage a group of services together.

Basic Settings

Name*
svg_cas_owa ?

Protocol*
SSL

Traffic Domain
+ /

Cache Type*
SERVER

AutoScale Mode
▼

☐ Cacheable
☒ State
☒ Health Monitoring
☒ AppFlow Logging

Number of Active Connections
0

Add backend servers as members of the service groups configured.

Create Service Group Member

☒ IP Based
 ☐ Server Based

IP Address/IP Address Range*

10

.

105

.

157

.

91

☐ IPv6

Port*

443

Weight

1

Server Id

None

Hash Id

0

☒ State

Create

Close

IP Address	Server Name	Port	Weight	Server Id	Hash Id	State
▶ 10.105.157.91	10.105.157.91	443	1	None	0	ENABLED
▶ 10.105.157.92	10.105.157.92	443	1	None	0	ENABLED
▶ 10.105.157.93	10.105.157.93	443	1	None	0	ENABLED

In this manner, the following service groups should be added. One service group is added for each protocol. When the servers are added correctly to each service group, their effective state will be **UP** as shown below:

NetScaler > Traffic Management > Load Balancing > Service Groups								
Service Group Name	State	Effective State	Protocol	Max Clients	Max Requests	Max Bandwidth (kbits)	Monitor Threshold	Traffic Domain
▶ svg_cas_owa	ENABLED	UP	SSL	0	0	0	0	0
▶ svg_cas_rpc	ENABLED	UP	SSL	0	0	0	0	0
▶ svg_cas_ews	ENABLED	UP	SSL	0	0	0	0	0
▶ svg_cas_autodiscovery	ENABLED	UP	SSL	0	0	0	0	0
▶ svg_cas_activesync	ENABLED	UP	SSL	0	0	0	0	0

Add custom monitors as shown below for each protocol. When creating the monitor, make sure the Secure option is enabled, as it is required for the monitor to successfully poll secure servers (this option is available in the Standard Parameters tab)

TOS ID Configuration:

TOS ID: [Text Field]

☒ Enabled
☐ Reverse
☐ Transparent
☐ LRTM (Least Response Time using Monitoring)
☒ Secure
☐ IP Tunnel

OK Close

Configure Monitor:

Name: mon_owa

Type: HTTP

Standard Parameters Special Parameters

HTTP Request: GET /owa/healthcheck.htm

☐ Treat Backslash as Escape Character

Response Codes: 200

Custom Header: [Text Field]

☐ Treat Backslash as Escape Character

OK Close

Add a monitor for each service you want to monitor using application-specific logic.

▶ mon_activesync	Enabled	HTTP
▶ mon_ews	Enabled	HTTP
▶ mon_owa	Enabled	HTTP
▶ mon_rpc	Enabled	HTTP

Bind the appropriate monitor to the service group.

Monitors			
Add Binding Edit Binding Unbind Edit Monitor			
Monitor Name	Weight	State	Passive
mon_owa	1	✓	✗
Close			

Add load balancing virtual servers

Add load balancing virtual servers as shown below. Set the IP address type to Non Addressable, as clients will not connect to this virtual server directly (In the server listing, this server will show an IP of 0.0.0.0)

The content switch virtual server will connect to this virtual server on the basis of content (or URL). The load balancing virtual server can be configured as HTTP or SSL, it will not make a difference for this deployment (this will matter when authentication is setup on the NetScaler appliance, this will be discussed in detail in a subsequent guide).

Note: If an SSL load balancing virtual server is setup, a valid SSL Certificate-Key combination will be required for successful configuration. This certificate can be added on the Basic Settings screen at any time, however the virtual server will be down until it is added.

Load Balancing Virtual Server

Basic Settings

Create a virtual server by specifying a name, an IP address, a port, and a protocol type. If an application is accessible only from the local area network (LAN) or wide area network (WAN) you can configure multiple virtual servers to receive client requests, thereby increasing the availability.

Name*

Protocol*

IP Address Type*

► More

OK Cancel

Bind service group to the virtual server. Bind services if a service group is not configured.

Load Balancing Virtual Server ServiceGroup Binding

Add Binding Unbind Edit Service Group

Service Group Name
 svg_cas_owa

Close

In the same way, add the following virtual servers, one for each protocol.

Add Edit Delete Enable Disable Statistics Action <div>Search</div>									
Filters: new_v_cas X Remove all									
Name	State	Effective State	IP Address	Port	Protocol	Method	Persistence	% Health	Traffic Domain
▶ new_v_cas_ews	Up	Up	0.0.0.0	0	SSL	LEASTCONNECTION	NONE	100.00% 1 UP/0 DOWN	0
▶ new_v_cas_owa	Up	Up	0.0.0.0	0	SSL	LEASTCONNECTION	NONE	100.00% 1 UP/0 DOWN	0
▶ new_v_cas_eas	Up	Up	0.0.0.0	0	SSL	LEASTCONNECTION	NONE	100.00% 1 UP/0 DOWN	0
▶ new_v_cas_oa	Up	Up	0.0.0.0	0	SSL	LEASTCONNECTION	NONE	100.00% 1 UP/0 DOWN	0
▶ new_v_cas_ecp	Up	Up	0.0.0.0	0	SSL	LEASTCONNECTION	NONE	100.00% 1 UP/0 DOWN	0

Content switch configuration

First create content switch actions that will be triggered if a content switching policy is hit. Select the virtual server to which the request should be directed, as shown below.

Create Content Switching Action

Name*

Target Load Balancing Virtual Server

☒ Name ☐ Expression

Target Load Balancing Virtual Server*

Comment

Add the following actions for each virtual server (or for each policy).

<input type="button" value="Add"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Rename"/> Search ▼					
Name	Target Load Balancing Virtual Server	Hits	Undefined Hits	Reference Count	Comment
new_cs_act_eas	Name : new_v_cas_eas	0	0	1	
new_cs_act_owa	Name : new_v_cas_owa	0	0	1	
new_cs_act_oa	Name : new_v_cas_oa	0	0	1	
new_cs_act_ews	Name : new_v_cas_ews	0	0	1	
new_cs_act_ecp	Name : new_v_cas_ecp	0	0	1	

Add policies as shown below for each virtual server. One policy and action is required for every virtual server.

Create Content Switching Policy

Name*

Action

Log Action

Domain

☒ Expression ☐ URL

Expression*

[Switch to Classic Syntax](#)

Add Edit Delete Action Search						
Name	Action	Log Action	URL	Expression	Domain	Hits
new_cs_pol_eas	new_cs_act_eas			HTTP.REQ.URL.CONTAINS("eas")		0
new_cs_pol_owa	new_cs_act_owa			HTTP.REQ.URL.CONTAINS("owa")		0
new_cs_pol_oa	new_cs_act_oa			HTTP.REQ.URL.CONTAINS("oa")		0
new_cs_pol_ews	new_cs_act_ews			HTTP.REQ.URL.CONTAINS("ews")		0
new_cs_pol_ecp	new_cs_act_ecp			HTTP.REQ.URL.CONTAINS("ecp")		0

Add the content switch virtual server as shown below.

Basic Settings

Name*

Protocol*

IP Address Type*

IP Address*
 ☐ IPv6

Port*

More

OK Cancel

Bind the policies to the content switch virtual server.

Content Switching Virtual Server Content Switching Policy Binding							
Add Binding Unbind Edit Search							
Priority	Policy Name	Expression	Action	Goto Expression	Invoke	Target Load Balancing Virtual Server	Hits
1	new_cs_pol_eas	HTTP.REQ.URL.CONTAINS("eas")	new_cs_act_eas				0
2	new_cs_pol_owa	HTTP.REQ.URL.CONTAINS("owa")	new_cs_act_owa				0
3	new_cs_pol_oa	HTTP.REQ.URL.CONTAINS("oa")	new_cs_act_oa				0
4	new_cs_pol_ews	HTTP.REQ.URL.CONTAINS("ews")	new_cs_act_ews				0
5	new_cs_pol_ecp	HTTP.REQ.URL.CONTAINS("ecp")	new_cs_act_ecp				0

Bind the server certificate to the content switch virtual server.

SSL Virtual Server Server Certificate Binding

Add Binding Unbind Update Certificate

Certificate Server Certificate for SNI

Close

Ensure that the content switch virtual server is up.

<div> Add Edit Delete Statistics Action </div>						Search ▾
Name	State	IP Address	Port	Protocol	Traffic Domain	
new_cs_cas_443	Up	192.168.1.70	443	SSL	0	

In the configuration shown above, a single namespace is used for all Exchange protocols. For example, for web access, the namespace is *https://mail.ctxns.net/owa*, and for Outlook clients the namespace is *https://mail.ctxns.net/oa*.

Similar configuration steps with domain-specific content switching policies will enable multiple namespace use case for all Exchange protocols where a client accesses a particular namespace for every service. For example, a web client uses the namespace *https://owa.mail.ctxns.net* and an Outlook client uses *https://oa.mail.ctxns.net*.

Conclusion

Citrix® NetScaler enables Microsoft Exchange 2013 deployment by ensuring remote access and load balancing of core components with intelligent monitoring. By serving as the front end, NetScaler can improve performance, scalability, availability and security of all Exchange 2013 deployments.

Corporate Headquarters
Fort Lauderdale, FL, USA

India Development Center
Bangalore, India

Latin America Headquarters
Coral Gables, FL, USA

Silicon Valley Headquarters
Santa Clara, CA, USA

Online Division Headquarters
Santa Barbara, CA, USA

UK Development Center
Chalfont, United Kingdom

EMEA Headquarters
Schaffhausen, Switzerland

Pacific Headquarters
Hong Kong, China



About Citrix

Citrix (NASDAQ:CTXS) is a leader in mobile workspaces, providing virtualization, mobility management, networking and cloud services to enable new ways to work better. Citrix solutions power business mobility through secure, personal workspaces that provide people with instant access to apps, desktops, data and communications on any device, over any network and cloud. This year Citrix is celebrating 25 years of innovation, making IT simpler and people more productive. With annual revenue in 2013 of \$2.9 billion, Citrix solutions are in use at more than 330,000 organizations and by over 100 million users globally. Learn more at www.citrix.com.

Copyright © 2015 Citrix Systems, Inc. All rights reserved. Citrix and NetScaler are trademarks of Citrix Systems, Inc. and/or one of its subsidiaries, and may be registered in the U.S. and other countries. Other product and company names mentioned herein may be trademarks of their respective companies.