Configuring Citrix NetScaler for IBM WebSphere Application Services

A deployment guide for configuring NetScaler load balancing and content switching
When deploying IBM WebSphere Application Servers, solution architects must balance cost against application requirements for high performance, security, and availability. As the number of users and applications increases, delivering a responsive application experience becomes a progressively greater challenge. Citrix NetScaler is an easy-to-install appliance that provides comprehensive load balancing, security, and offloading features for IBM WebSphere deployments. Deploying NetScaler helps to optimize application performance, security, and availability, helping to improve the overall user experience.

**Why NetScaler for IBM WebSphere workloads**

NetScaler App Delivery Controller™ helps to improve application delivery for IBM® WebSphere™ Application Servers—even under heavy load conditions. Citrix® NetScaler® appliances combine advanced application-based functionality with capabilities of traditional load balancers, traffic managers, and remote access systems.

For WebSphere application workloads, NetScaler appliances control and direct incoming web-based client requests (Figure 1). Based on customized policies and WebSphere server availability, NetScalers intelligently distribute traffic to available WebSphere Application Servers.

![NetScaler appliances direct application traffic to IBM WebSphere Application Servers.](image)

**NetScaler features include:**

- Comprehensive and centralized policy management.
- Layer 4 (TCP and UDP) through Layer 7 (FTP, HTTP, and HTTPS) traffic management and load balancing.
- Performance and optimization features (e.g., keep-alive, compression, caching, buffering) that reduce transaction times and increase application responsiveness.
• Security features (e.g., firewall; authentication/authorization/accounting (AAA); filtering; and denial of service (DoS) protection).
• Availability features (e.g., detecting unavailable servers and directing application requests to the remaining servers).
• Visibility. NetScaler allows centralized and efficient system configuration and management of traffic, events, and performance. NetScaler uses an intuitive policy builder to create application delivery policies without a need for writing complex programs or scripts.

Example NetScaler deployment scenarios for IBM WebSphere
This paper describes how to configure two of the most commonly used NetScaler capabilities for IBM WebSphere workloads: load balancing and content switching. The paper covers two deployment scenarios:

• Scenario #1: Load balancing application traffic across a pool of IBM WebSphere Application Servers. Based on configured policies, load balancing distributes application requests across multiple IBM WebSphere servers hosting the application. As a result, NetScaler deployments can minimize the possibility of overtaxing a single IBM WebSphere server and causing unresponsive application behavior. Load balancing also facilitates fault tolerance. If an application server becomes unavailable, NetScaler directs traffic to another server that hosts the same application.
• Scenario #2: Content switching in addition to load balancing. Content switching allows traffic to be directed to different servers based on application-specific criteria. For example, requests can be directed to different servers to support specific language requirements, geographic proximity, etc.

This deployment guide describes how to configure NetScaler to perform load balancing for WebSphere applications. It then extends the load-balancing scenario, adding the ability to do content switching. While this guide doesn’t cover how to implement other NetScaler capabilities, NetScaler appliances can be configured for SSL Offloading, Layer 7 Application Firewall functionality, and Global Sever Load Balancing (GSLB). GSLB extends load balancing across the end-to-end enterprise, distributing IBM WebSphere traffic across multiple datacenters based on proximity, load, or availability. In this way, NetScalers can improve response time and support disaster recovery scenarios.

Assumptions
The procedure in this guide describes how to configure NetScaler VPX™, the virtual server implementation of NetScaler technology. Roughly the same procedures can be used to configure load balancing and content switching on physical NetScaler models, however. For the purpose of this paper, an engineer installed and configured NetScaler and IBM WebSphere software in a test lab environment to validate the configuration procedures. The lab implementation used NetScaler 10.5 and IBM WebSphere 8.5.5.0 software releases.

The procedure here assumes that the IBM WebSphere and Citrix NetScaler software installations have been previously completed and focuses on port-installation configuration tasks. (See the Citrix NetScaler documentation—Installing the NetScaler hardware or Getting Started with Citrix NetScaler VPX—for software installation guidance.) The procedure also assumes that NetScaler licensing has been properly installed.
The default IBM WebSphere configuration includes an application server and an enterprise application called Default Application. Default Application includes several servlets that you can use to verify that the WebSphere Application Server is operational. To validate NetScaler configurations in a typical deployment scenario, the test environment used three application servers and two Default Application servlets (Snoop and HitCount) on port numbers 9080 and 9081 respectively. Before configuring NetScaler functionality, the WebSphere Application Servers must be accessible via known IP addresses.

**Initializing NetScaler**

Before configuring NetScaler for different deployment scenarios, enable Basic Features on the NetScaler as follows.

### Instructions

Log into the NetScaler.

### Visual

Under the System > Settings menu, select Configure Basic Features.
Scenario #1: Configuring NetScaler to load balance WebSphere workloads
The first deployment scenario configures NetScaler to load-balance application traffic across the pool of IBM WebSphere Application Servers. For this scenario, NetScaler uses a load-balancing virtual server for each application. The virtual server proxies client requests, balancing application load among IBM WebSphere servers in the farm. As shown in Figure 2, the NetScaler is configured with two virtual servers, LB VIP 1 and LB VIP 2, for load balancing of the two applications.

Figure 2: Scenario #1- Configuring NetScaler to load balance IBM WebSphere traffic (logical view).

The table below summarizes configuration details for the first deployment scenario.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
</table>
| Applications                                  | Port 9080: Application 1 (snoop)  
Port 9081: Application 2 (HitCount) |
| IBM WebSphere Application Servers             | Server 1, IP 172.16.0.81  
Server 2, IP 172.16.0.82  
Server 3, IP 172.16.0.83 |
| Load Balancing service names                  | For the first application: app1-81, app1-82, app1-83  
For the second application: app2-81, app2-82, app2-83 |
| Load-Balancing Virtual Server names & IPs     | LB VIP 1: 172.16.0.96  
LB VIP 2: 172.16.0.97 |
End users specify a virtual server for the application, giving the load-balancing virtual IP address (LB VIP 1 or LB VIP 2). For example, users enter a URL such as http://172.16.0.96/snoop or http://172.16.0.97/HitCount. Each LB VIP is bound to application services associated with the pool of WebSphere servers. Based on the load-balancing algorithm, NetScaler routes the request to an available application server in the pool.

Configuring load balancing for IBM WebSphere Application Servers

There are four basic steps to configure NetScaler for load balancing: (1) define the pool of IBM WebSphere Application Servers, (2) define load-balancing services associated with the application, (3) create a virtual server to process requests, and (4) bind load-balancing services to the virtual server. Step-by-step instructions with screenshots follow.

**Instructions**

Before you begin, confirm that the NetScaler Load Balancing feature is enabled under Traffic Management. (No yellow warning symbol is present, which would indicate that the feature is disabled.)

Expand the Load Balancing menu.

Select Servers from the Load Balancing menu.
On the Create Server page, add a server definition for each of the IBM WebSphere Application Servers that will be used in load balancing, and click Create.

In this example, repeat this step to add three servers: 172.16.0.81, 172.16.0.82, and 172.16.0.83.

Select Services from the Load Balancing menu.

Click Add.
Define the load-balancing services associated with a port on the WebSphere servers. For example, the service app1-81 is configured to use server 172.16.0.81 on port 9080. Click OK to complete the service definition.

Repeat to configure services app1-82 and app1-83 (for application servers 172.16.0.82 and 172.16.0.83 respectively). These services also use HTTP protocol and port 9080.

The Services menu shows the configured services and their state. “Up” indicates that the specified port is open on the corresponding server and able to communicate.

Select Virtual Servers from the Load Balancing menu to define a virtual server for the first application.

Enter the settings for the load-balancing virtual server. In this example, the virtual server for the first application is named app1-vip. Specify the load-balancing virtual IP address (LB VIP 1, 172.16.0.96) and the port number on which users will make client requests (80, the default HTTP port).
At this point, the virtual server is configured but there are no services bound to it.

Click on “No Load Balancing Virtual Server Service Binding” to create a service binding.

Expand the Select Service button and select the service to be bound (app1-81).

Click Bind to bind the service to the virtual server.

Click on Add Binding and repeat the process to bind the other services (app1-82 and app1-83) to this virtual server.

When all three services for the application are bound to the virtual server, click Close.
The Virtual Server screen now shows three configured bindings.

Click Done.

The virtual server configuration is complete.

The default load-balancing algorithm is Least Connection, which directs the request to the service that currently has the fewest client connections.

Test NetScaler load balancing. In a browser, enter the LB VIP of the virtual server (172.16.0.96) and the name of application (snoop).

A side benefit of using snoop as a test application is that it displays the IP of the IBM WebSphere server on which the application executes. NetScaler directs subsequent client requests to different servers to balance the load.
NetScaler offers excellent visibility into load balancing behavior across the IBM WebSphere servers. On the NetScaler Dashboard, you can view load-balancing statistics by expanding the virtual server app1-vip. In the example results below, five requests for the first application have been distributed evenly across three different WebSphere servers.

**Configuring load balancing for a second application**

By repeating the load-balancing configuration process—specifying the IBM WebSphere servers to be used, defining load-balancing services, creating a virtual server for the application, and binding services to the virtual server—it’s easy to configure load balancing for additional applications. The following steps configure load balancing for a second application (the servlet HitCount) on the same set of IBM WebSphere Application Servers.

**Instructions**

To configure services for the second application, use the Load Balancing > Services menu to configure new service entries.

For example, define the services app2-81, app2-82, and app2-83. These services use port 9081 on the same set of servers (172.16.0.81, 172.16.0.82, and 172.16.0.83 respectively).
Under Load Balancing > Virtual Servers, create the virtual server for the second application, app2-vip.

Click OK.

Bind services to the virtual server app2-vip.

Click the radio button for the service (e.g., app2-81) and click OK to bind it to the virtual server. Repeat to bind other services for the second application (app2-82 and app2-83). After all the bindings are configured, click Done.

Test NetScaler load balancing for application requests.

In a browser, enter the LB VIP for the second application (LB VIP 2, 172.16.0.97) and the servlet name (HitCount).
**Scenario #2: Configuring NetScaler for IBM WebSphere load balancing and content switching**

Figure 3 illustrates the second deployment scenario. This scenario builds on the previous one, adding content switching in addition to load balancing. Content switching allows client requests to be directed to load-balancing virtual servers (and in turn to IBM WebSphere Application Servers) based on rules configured via NetScaler policies and actions.

In the test environment, we configured a content switching virtual server with an IP address of 172.16.0.98. To access either application (snoop or HitCount), client requests include this URL. Based on rules for matching URL substrings (e.g., the application name “snoop” or HitCount”), NetScaler directs the client request to the appropriate LB VIP: LB VIP 1 for application 1 (snoop) or LB VIP 2 for application 2 (HitCount). NetScaler then load balances application requests across the associated pool of application servers.

![Figure 3: Deployment Scenario #2- Configuring NetScaler to content switch and load balance IBM WebSphere traffic (logical view).](image)

The table below summarizes configuration details for the second deployment scenario. The shaded rows indicate settings defined previously for load balancing.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applications</td>
<td>Port 9080: Application 1 (snoop)</td>
</tr>
<tr>
<td></td>
<td>Port 9081: Application 2 (HitCount)</td>
</tr>
<tr>
<td>IBM WebSphere Application Servers</td>
<td>Server 1, IP 172.16.0.81</td>
</tr>
<tr>
<td></td>
<td>Server 2, IP 172.16.0.82</td>
</tr>
<tr>
<td></td>
<td>Server 3, IP 172.16.0.83</td>
</tr>
<tr>
<td>Load Balancing service names</td>
<td>For the first application: app1-81, app1-82, app1-83</td>
</tr>
<tr>
<td></td>
<td>For the second application: app2-81, app2-82, app2-83</td>
</tr>
<tr>
<td>Load-Balancing Virtual Server names</td>
<td>LB VIP 1 for the first application: app1-vip, 172.16.0.96</td>
</tr>
<tr>
<td></td>
<td>LB VIP 2 for the second application: app2-vip, 172.16.0.97</td>
</tr>
<tr>
<td>Content Switching Action names</td>
<td>For the first application: app1-action</td>
</tr>
<tr>
<td></td>
<td>For the second application: app2-action</td>
</tr>
</tbody>
</table>
Content Switching Policy names and expressions

- For the first application: app1-policy contains the expression HTTP.REQ.URL.CONTAINS("/snoop")
- For the second application: app2-policy contains the expression HTTP.REQ.URL.CONTAINS("/HitCount")

Content Switching Virtual Server name and VIP

CS VIP for both applications: app1-csvip, 172.16.0.98

Configuring content switching in addition to load balancing

There are four basic steps to extend the previous scenario and configure NetScaler for content switching in addition to load balancing: (1) define a content switching action for the application, (2) define a policy that contains rules for when to apply the action, (3) create a content switching virtual server to process client requests, and (4) bind the content switching policies to the virtual server. Step-by-step instructions follow for setting up NetScaler content switching.

Instructions

Before you begin, confirm that the NetScaler Content Switching feature is enabled under Traffic Management.

Expand the Content Switching menu.

Under Content Switching > Actions, define an action.

For the first application, define an action called app1-action. The action directs application requests to the target load-balancing virtual server, in this case app1-vip.

Click Create.
Under Content Switching > Policies, define a policy.

Enter a policy name, the action to perform when the expression is true, and the expression that controls content switching.

In this example, the action is applied when the substring “snoop” appears in the URL.

Click Create.

Under Content Switching > Virtual Server, define settings for the content switching virtual server app1-csvip.

Enter the IP address for the virtual server (the CS VIP). 172.16.0.98.

Click OK.
At this point, the content switching virtual server is configured but there is no policy bound to it.

Click on “No Content Switching Policy Bound” to bind a policy to the virtual server.

Select the policy to bind to the virtual server and enter a priority for the binding.

For example, bind the policy created for the first application, app1-policy, to the content switching virtual server.

Click Bind.

Review the virtual server configuration and click OK.
Repeat the steps to define a content-switching action and policy for the second application. In this case the pattern-matching expression in the policy looks for a substring match for HitCount.

Under Content Switching > Virtual Servers, bind the policy for the second application (app2-policy) to the content-switching virtual server (app1-csvip).

Click Bind.

Test NetScaler content switching.

Specify URLs with the content-switching virtual server (the CS VIP) and application name.
Refresh the dashboard. NetScaler updates the number of hits against each content-switching policy. Additional metrics are visible under the NetScaler Dashboard tab.

**Summary**

As shown here, it’s easy to configure NetScaler appliances for load balancing and content switching traffic to IBM WebSphere Application Servers. NetScaler supports a number of load balancing algorithms beyond the default Least Connection method implemented in these examples. Other load balancing schemes include round robin; least response time, bandwidth, or packets; hashing based on URL, domain name, source IP and/or destination IP address; and tokens. As shown here, content switching is a simple extension of load balancing, directing client requests to load-balancing virtual servers. NetScaler can apply content switching to control HTTP, HTTPS, FTP, TCP, Secure TCP, and RTSP client requests.

By load balancing and content switching across IBM WebSphere Application Servers, NetScaler can improve both application availability and responsiveness. NetScaler also enables excellent visibility into how traffic is being directed, making it easier to manage application workloads and deliver a positive end user experience.

Administrators can further extend the deployment scenario presented in this guide to incorporate other NetScaler features for IBM WebSphere workloads. Global Server Load Balancing (GSLB), for example, distributes traffic across datacenters, creating a highly resilient architecture that supports disaster recovery. NetScalers can use various criteria for GSLB distribution, including least connection, static proximity, or dynamic proximity. When GSLB is implemented for IBM WebSphere servers, if a link to a datacenter goes down, NetScaler can redirect traffic to an available datacenter.

**Learn more**

For more information about deploying Citrix NetScaler, see the following resources.

Citrix NetScaler site
http://www.citrix.com/netscaler

Citrix NetScaler documentation
http://support.citrix.com/proddocs/topic/netscaler/ns-gen-netscaler-wrapper-con.html
Citrix (NASDAQ:CTXS) is leading the transition to software-defining the workplace, uniting virtualization, mobility management, networking and SaaS solutions to enable new ways for businesses and people to work better. Citrix solutions power business mobility through secure, mobile workspaces that provide people with instant access to apps, desktops, data and communications on any device, over any network and cloud. With annual revenue in 2014 of $3.14 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

About Citrix

Copyright © 2015 Citrix Systems, Inc. All rights reserved. Citrix, CloudBridge, NetScaler App Delivery Controller, NetScaler and Citrix Ready 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.