Citrix NetScaler 10 Essentials and Networking

Citrix Course CNS-205-1I

© Copyright 2012 Citrix Systems, Inc.

Citrix NetScaler 10 Essentials and Networking

Citrix Course CNS-205-1I
October 2012
Version 1.0

Citrix® authorized
Courseware

© Copyright 2012 Citrix Systems, Inc.

Table of Contents

Module 1: Getting Started 23
Overview 25
NetScaler Overview 26
Introduction to the NetScaler System 27
NetScaler Functionality 28
NetScaler Operating System Overview 29
Hardware Platforms 30
Hardware Compliance 31
Hardware Components 32
  Network and Serial Interfaces 32
  Redundant Power Supplies 32
  File System 32
Planning a NetScaler Deployment 33
  One-Arm Mode 33
  Two-Arm Mode 34
Test Your Knowledge 36
Deployment Scenarios 37
  Flex-Tenancy 37
  Multi-Tenancy 38
  Displacement 38
  TriScale 39
NetScaler Configuration 41
Logging in to the NetScaler System 42
NetScaler Licenses 43
  To Install the NetScaler License 43
Test Your Knowledge 45

Module 2: Basic Networking 47
Overview 49
OSI Networking Model 50
NetScaler Architecture Overview 51
Packet Forwarding 52
  Layer-2 Mode 52
  Layer-3 Mode 53
  MAC-based Forwarding Mode 53
NetScaler-owned IP Addresses 54
  NetScaler IP Address 54
  Subnet IP Address 55
  Mapped IP Address 56
  Virtual IP Address 57
  DNS Virtual IP Addresses 57

© Copyright 2012 Citrix Systems, Inc.

To Remove a NetScaler-Owned IP Address 57
NetScaler Network Interfaces 60
Enable or Disable Network Interfaces 60
SNIP Address Binding to Interfaces 60
MAC-based Forwarding Mode 62
NetScaler Connection Proxying 63
Determining the Destination IP Address 64
Determining the Source IP Address 65
Specified Source IP Addresses for Back-end Communication Using Network Profiles 65
Sending a Client IP Address to Servers 66
Use Source IP Mode 67
Configuring USIP Mode 69
Virtual Local Area Networks (VLAN) 70
VLAN Configuration 71
NSVLAN 72
Port-based VLANs 72
Default VLAN 72
Tagged VLANs 73
Configuring VLANs 73
Network Address Translation (NAT) 75
Inbound Network Address Translation 75
Coexistence of INAT and Virtual Servers 75
Reverse Network Address Translation (RNAT) 77
Test Your Knowledge 78

Module 3: Basic Load Balancing 79
Overview 81
Load-Balancing Basics 82
Entity Management 83
Traffic Types 84
Configure Basic Load Balancing 86
Enable the Load-Balancing Feature 86
To Enable Load Balancing Using the Configuration Utility 86
Load Balancing Enablement Using the Command-Line Interface 86
Server Creation 87
Server Configuration Parameters 87
To Create a Server Using the Configuration Utility 88
Server Creation using the Command-Line Interface 88
Services Configuration Overview 89
Services Creation 89
Service Configuration Parameters 90
To Create a Service Using the Configuration Utility 91
Service Creation Using the Command-Line Interface 91
Virtual Server Creation 92
Virtual Server Configuration Parameters 92
To Create a Virtual Server Using the Configuration Utility 93

© Copyright 2012 Citrix Systems, Inc.

Virtual Server Creation Using the Command-Line Interface 93
Services Bound to a Virtual Server 94
To Bind a Service to a Load-Balancing Virtual Server by Using the Configuration Utility 94
Services Bound to a Virtual Server Using the Command-Line Interface 94
Disabling Services 95
To Disable a Virtual Server Using the Configuration Utility 95
Service Disablement using the Command-Line Interface 95
Removing Services 96
To Remove a Service Using the Configuration Utility 96
Removal of a Service using the Command-Line Interface 96
Configuration Verification 97
Test Your Knowledge 99
Service Level Monitors 101
Default Monitors 102
Built-in Monitors 102
Custom Monitors 104
Customization of Load-Balancing Configuration 105
Load-Balancing Methods 105
Service Weights 108
To Configure a Virtual Server to Assign Weights to Services Using the Configuration Utility 108
Virtual Server Configuration to Assign Weights to Services Using the Command-Line Interface 108
Persistence and Persistence Connections 109
Session Persistence Methods 109
Test Your Knowledge 113

Module 4: High Availability 115
Overview 117
High Availability Functionality 118
High Availability Node Configuration 120
Pre-Configuration Checklist 120
Virtual Media Access Control Address 120
Configuring Primary and Secondary Nodes 121
High-Availability Status 121
Propagation and Synchronization 123
Disabling Command Propagation 123
Automatic Configuration Synchronization 124
Forced Synchronization 124
Performing a Forced Failover 125
High Availability Management 126
Upgrading a High Availability Pair 126
Test Your Knowledge 127

Module 5: Policies and Expressions 129

© Copyright 2012 Citrix Systems, Inc.

Overview 131
Policies Overview 132
Policy Basics 132
Basic Policy Components 132
About Policy Bindings 133
Policy Priorities 134
Hypertext Transfer Protocol 134
Expression Structures 136
Wildcards 136
Context-Sensitive Fields 136
Simple Expressions 137
Compound Expressions 137
Advanced Policy Conversion 139
Test Your Knowledge 140

Module 6: Content Switching 141
Overview 143
Introduction to Content Switching 144
Understanding Content Switching 145
Content Switching Based on Layer Characteristics 146
Content-Switching Virtual Servers and Load-Balancing Virtual Servers 147
Configuring Content-Switching Virtual Servers 148
Test Your Knowledge 150
Configure a Load-Balancing Setup for Content Switching 151
Content-Switching Policies 152
Binding Content-Switching Policies 153
Content-Switching Rule Precedence Without Priority Specified 155
Content-Switching Rule Precedence With Priority Specified 156
Unmatched Traffic Handling 157
Test Your Knowledge 158
Test Your Knowledge 159

Module 7: Connection Tuning 161
Overview 163
IPv6 164
Load Balancing in Direct Server Return Mode for IPv6 Networks 164
VLAN as Gateway in the IPv6 Routes 165
Reverse Network Address Translation for IPv6 Traffic 165
DNS and the NetScaler 167
NetScaler as an ADNS Server 168
NetScaler as a DNS Proxy Server 169
NetScaler as a DNS End Resolver 169
NetScaler as a DNS Forwarder 170
IP Routing 171
Static Routes 171

© Copyright 2012 Citrix Systems, Inc.

Policy-based Routing 171
Policy-based Routing Parameters 172
Policy-based Routes for IPv6 Traffic 173
Test Your Knowledge 175
Configuration of Load Balancing for Commonly-Used Protocols 176
Link Load Balancing 177
IPv6 Support for Link Load Balancing 177
SIP Load Balancing 178
Advanced Load-Balancing Methods 179
Least-Response-Time 179
Token 179
Token Method for Load-Balancing DataStream Example 180
Token Method for Load-Balancing Architecture 180
Hashing 180
The URL Hash Method 181
The Domain Hash Method 182
The Destination IP Hash Method 182
The Source IP Hash Method 182
The Source IP Destination IP Hash Method 182
The Source IP Source Port Hash Method 182
The Call ID Hash Method 183
Custom Load 184
HTTP-ECV and TCP-ECV Monitors 185
Persistence and Persistence Connections 186
Session Persistence Method Overview 186
Persistence Tables 188
Persistence Group Configuration 188
Viewing of Persistence Sessions 189
Clearing of Persistence Sessions 189
Test Your Knowledge 191
Load-Balancing Configuration Protection 192
Backup Load-Balancing Virtual Server Configuration 192
Diverting Excess Traffic to a Backup Virtual Server 192
Spillover Configuration 192
Failover Configuration 193
Test Your Knowledge 194
Load-Balancing Setup Management 196
Services Management 196
Advanced Services Options 196
Service Groups 199
Virtual Servers Management 200
The Load-Balancing Visualizer 200
Traffic Management 202
Configuration of Sessionless Load-Balancing Virtual Servers 202
Direct Request According to Priority 202
Delayed Cleanup of Virtual Server Connections Enablement 202
Graceful Shutdown of Services 203

© Copyright 2012 Citrix Systems, Inc.

Use of a Specified Source IP for Back-end Communications 203
Enabling of TCP Buffering of Responses 204
Maintaining of Client Connection for Multiple Client Requests 204
Use of the Source IP Address of the Client When Connecting to the Server 205
Test Your Knowledge 206

Module 8: Global Server Load Balancing 209
Overview 211
GSLB Deployment Methods 212
GSLB Concepts 213
GSLB Conversation Process 214
GSLB Entities 216
GSLB Entity Relationship 216
GSLB DNS Methods 218
Name Servers 218
External DNS Server Example 219
DNS Proxy Configuration 220
DNS Request Scenarios 220
Resource Records 221
Authoritative DNS Service 222
Viewing and Configuring GSLB with the Visualizer 223
Implementing Static GSLB 224
ADNS Service Configuration 224
Basic GSLB Site Configuration 225
GSLB Service Configuration 225
Services Management 225
GSLB Virtual Server Configuration 226
Binding of GSLB Services to a GSLB Virtual Server 226
Binding of a Domain to a GSLB Virtual Server 226
Synchronizing a GSLB Configuration 227
GSLB with Sub-Domains 227
Metric Exchange Protocol 228
Metric Information Types 228
To Configure Network Metric Exchange 228
To Configure Site Metric Exchange 229
To Configure Persistence Information Exchange 229
GSLB Monitoring Configuration 230
Test Your Knowledge 231
Configuring Site-to-Site Communication 232
RPC Node Password 233
Encryption of Site Metrics Exchange 234
Source IP Address for an RPC Node 235
GSLB Site Communication Example 236
Customizing the GSLB Configuration 237
Modifying Maximum Connections or Minimum Bandwidth for a GSLB Service 237
Creating CNAME-based GSLB Services 237

© Copyright 2012 Citrix Systems, Inc.

Changing the GSLB Method 239
Proximity-based Load-Balancing Methods 239
Implementing Proximity-based GSLB 240
Static Proximity Configuration 241
Dynamic RTT Configuration 241
Static and Dynamic RTT Configuration 242
GSLB Persistence 243
Configuring of Persistence Connections 243
Configuring of Persistence-based on Source IP 244
Persistence-based on HTTP Cookies 244
Using Dynamic Weights for Services 246
Monitoring GSLB Services 247
Adding Monitors 247
Binding Monitors to a GSLB Service 248
To Bind a Monitor 248
Monitoring GSLB Sites 249
Protecting the GSLB Setup Against Failure 250
Backup GSLB Virtual Server Configuration 250
Configuration of a GSLB Setup to Respond to Multiple IP Addresses 250
Responding with an Empty Address Record When DOWN 251
Backup IP Address Configuration for a GSLB Domain 252
Diverting Excess Traffic to a Backup Virtual Server 253
Commonly-Used Deployment Scenarios for GSLB 255
Implementing GSLB Failover for Disaster Recovery 256
GSLB in an Active-Standby Datacenter Setup 256
GSLB in an Active-Active Datacenter Setup 256
GSLB with Weighted Round Robin 257
GSLB with Datacenter Persistence 257
Test Your Knowledge 259

Module 9: Clustering 261
Overview 263
How Clustering Works 264
Cluster Synchronization 265
Test Your Knowledge 266
Cluster Connections 267
Cluster Communication Interfaces 268
Striped and Spotted IP Addresses 269
Traffic Distribution 270
Cluster and Node States 271
NetScaler Cluster Setup 272
To Set Up a Cluster Backplane 272
To Add the Initial Configuration Coordinator 272
To Add the Initial Configuration Coordinator (Configuration Utility) 273
Cluster Node Addition 274
To Add a Node to the Cluster 274

© Copyright 2012 Citrix Systems, Inc.

NetScaler Cluster Traffic Distribution Mechanisms 275
To Configure Cluster Link Aggregation Group Statically 276
Dynamic Configuration for a Cluster Link Aggregation Group 277
Test Your Knowledge 278
NetScaler Cluster Management 279
Removing a Cluster Node 279
To Disable a Cluster Node 279
To View Cluster Instances and Node Details 280
To Synchronize Files on a Cluster 280
Cluster Nodes Software Upgrades and Downgrades 281
NetScaler Cluster Troubleshooting 282
To Trace the Packets of a Netscaler Cluster 282
Troubleshooting Joining and Syncing Issues in a Cluster 283
Troubleshooting Common Issues 284
Test Your Knowledge 286

Module 10: Security and Authentication 287
Overview 289
SSL 290
SSL Administration 290
SSL Session Process 291
SSL Keys 292
Public Key Encryption 293
Digital Certificates 294
Certificate Signing Request 294
SSL Certificates 295
Certificate Generation 296
Certificate Key Pairs 296
Certificate Revocation List (CRL) 298
Certificate Updates 298
NetScaler System Communication 300
Deployment Scenarios 302
Front-end SSL with Back-end HTTP Requirements 302
Securing Traffic Between a Client and the NetScaler System 303
Front-end SSL with Back-end SSL Requirements 304
Securing Traffic Between the NetScaler and the Server 304
Front-end SSL_TCP with Back-end TCP Requirements 305
Securing TCP Traffic Between a Client and the NetScaler System 305
SSL_Bridge 306
SSL_Bridge Requirements 306
SSL Termination Points 307
Features and Benefits 308
Test Your Knowledge 309
SSL Offload Overview 310
Configuring SSL Offload 310
Offload Performance 311

© Copyright 2012 Citrix Systems, Inc.

Advanced SSL Settings 313
SSL Redirect 313
SSLv2 Protocol and Redirect 313
SSL Citrix Recommendations 314
Creating SSL Virtual Servers 315
Binding an SSL Virtual Server 315
Cipher Suite Binding to an SSL Virtual Server 315
Test Your Knowledge 317
Authentication Setup 318
AAA for Traffic Management 318
SAML 2.0 Consumer Support 318
Authentication Policy Creation 320
Authentication Policy Binding 320
Resource Identification for Users 321
Single Sign-on Setup 322
LDAP Authentication 323
Test Your Knowledge 324
Access Control Lists 325
Access Control Lists Precedence 325
Matching Access Control List Entries 325
Access Control List Configuration 326
Simple Access-Control-List Configuration (Command-line Interface) 327
Detailed Access Control Lists Configuration 328
Extended Access Control Lists 328
Creation and Removal of Access Control Lists 328
Access Control Lists Application 329
Viewing of Access Control Lists 330
Removal of Access Control Lists from the System 330
Access-Control-List Example 330
IPv6-based Access-Control-List Support 331
Core Protection Enhancements 332
SYN Protection 333
SSL Renegotiation Attack 334
TLS Man-in-the-Middle Attacks 335
Adaptive Request Timeout for HTTP DoS 336
Slow-Header, Slow-Post, and Slow-Read Attacks 337
Client Traffic Management Based on Traffic Rate 339
Test Your Knowledge 340

Module 11: Responder, Rewrite, and URL Transform 341
Overview 343
Understanding Packet-Processing Flow 344
Understanding Policies 346
Policy-Process-Evaluation Flow 347
Identifying Advanced Policy Expressions 348
Advanced-Policy-Expression Syntax 348

© Copyright 2012 Citrix Systems, Inc.

Actions 350
Action Syntax 350
Understanding Bind Points 351
Policy Binding Evaluation Process 351
Binding in the Policy Manager 352
Understanding Policy Labels 353
Configuring Policy Labels 353
Pattern Sets 354
How String Matching with a Pattern Set Works 354
Configuring a Pattern Set 354
Using Pattern Sets 355
Pattern Set Operators 355
Typecasting 357
Test Your Knowledge 359
Rewrite, Responder, and URL Transformation 361
Rewrite Process 362
Responder Process 363
URL Transformation Process 364
Transformed Elements 365
Configuring Policies and Actions 366
Rewrite Default Actions 367
Configuring a Rewrite Action 368
Rewrite Action Parameters 370
To Configure a Rewrite Action Using the Configuration Utility 371
Create a Rewrite Action Using the Command-line Interface 372
Rewrite Policies 373
Responder Default Actions 374
Responder Actions 375
Respondwith 376
To Create a Respondwith Action 376
Responder Action for Timeouts 377
Responder Policies 378
Responder HTML Page Imports 379
Binding Policies 380
Configuring URL Transformation 381
URL Transformation Actions 382
Binding of URL Transformations 384
Test Your Knowledge 385

Module 12: Optimizing Traffic 387
Overview 389
Introduction to Compression 390
Compression Process 390
Compression Considerations 390
Compression Responses 391
Compression Parameters 391

© Copyright 2012 Citrix Systems, Inc.

Compression Policies 392
Setting Compression Policies Globally 393
Configuring Compression 393
Compression Actions 394
Integrated Caching 395
Reverse-Proxy-Cache Configuration 396
Content Groups 397
Cache Selectors and Policies 398
Caching Static and Dynamic Content 399
Dynamic Content 399
Dynamic Content Example 399
Static Content Example 399
Cache Actions 399
Request and Response Process Flow 401
Request Side 401
Cache Policies and Cache Expressions 403
Cache Policies 403
Binding Cache Policies 404
Cache Policy Evaluation 404
Action Analytics 406
Evaluation Order 407
Built-in Policies 408
Graceful Cache Configuration Changes 409
Test Your Knowledge 410
Cache Content Groups 411
Reversed Content Group Names 411
Cache Content Aging 411
Content Group Settings 412
HTTP Headers and Content Group Settings 413
Other Content Group Attributes 413
Flash Cache 414
Global Cache Attributes 415
Cache Arguments 416
verifyUsing Argument 417
HOSTNAME Setting 418
Caching Management 419
AppExpert Templates 420
Using AppExpert Templates 420
Using the Application Interface 420
Applications Pane 421
Application Units 422
AppExpert Template Scalability 422
Planning for Applications 422
Terminology 423
Methodology 424
Application Example: Part 1 425
Application Example: Part 2 426

© Copyright 2012 Citrix Systems, Inc.

AppExpert Template Deployment 427
Test Your Knowledge 428

Module 13: Monitoring 429
Overview 431
Monitoring Needs 432
Monitoring Methods 433
SNMP 433
AppFlow 433
EdgeSight Monitoring 433
Action Analytics 434
Command Center 434
Reporting Tool 435
Working with Reports 435
Using Built-in Reports 436
Dashboard 436
System and Feature Counters 436
Dashboard Components 437
NetScaler Log Management 439
Audit Logging 439
Configuring NetScaler for Audit Logging 439
Configuring an Auditing Server 440
Global Auditing Parameters 441
Configuring Auditing Policies 441
Binding Auditing Policy 442
Audit Messages 442
Test Your Knowledge 443
Simple Network Management Protocol 444
SNMPv3 444
SNMPv3 Entities 445
Managed Devices 446
SNMP Agents 446
SNMP Management Systems 446
Management Information Base 447
Community Strings 447
Traps 447
SNMPv1 and SNMPv2 Communication Process 448
To Configure SNMPv1 and SNMPv2 450
Configuring SNMP Management Systems 450
Configuring MIB System Variables 451
Configuring Community Strings 451
Configuring Alarms 452
Configuring SNMP Traps 452
Test Your Knowledge 454
AppFlow on the NetScaler System 455
How AppFlow Works 455

© Copyright 2012 Citrix Systems, Inc.

Configuring AppFlow 455
Specifying an AppFlow Collector 456
AppFlow Action Configuration 456
AppFlow Policy Configuration 457
AppFlow Policy Binding 457
AppFlow for Virtual Servers 458
AppFlow Collectors and EdgeSight Monitoring 459
Prerequisites for Exporting Performance Data to AppFlow Collectors 460
AppFlow Actions and EdgeSight Monitoring Responder Policies 461
Third-Party Collectors 462
Test Your Knowledge 463

Module 14: Troubleshooting 465
Overview 467
Troubleshooting Methodology 468
Define the Issue 469
Gather Detailed Information 469
Additional Resources 469
Narrow the Scope 470
Consider Possible Causes 471
Create an Action Plan 471
Implement the Action Plan 471
Observe the Results of the Action Plan 472
Document Changes Following Issue Resolution 472
Troubleshooting Resources 474
Citrix Knowledge Center 474
Citrix Product Documentation 474
Citrix Technical Support 474
Collected NetScaler Data 474
Troubleshooting Log 475
Troubleshooting Tools 476
nsconmsg 476
newnslog 477
Comparative Configuration 477
Show TechSupport Command 478
Using the Visualizer 479
Test Your Knowledge 480
Networking 481
Duplex Mismatch or Misconfigured Interface Settings 481
Inaccessible Content 482
Testing of Network Connectivity Through a Firewall 482
Display NetScaler System Information 483
Troubleshooting Hardware Issues 483
Non-Maskable Interrupt (NMI) Button 484
Display Software Information 484
Troubleshooting License Issues 485

© Copyright 2012 Citrix Systems, Inc.

Load-Balancing Issues 486
Uneven Load Balancing 486
Service or Virtual Server Flapping 488
Load Balancing of Third-Party Servers 488
Authentication, Authorization, and Access (AAA) Issues 489
Authentication Troubleshooting 489
SSL Troubleshooting 490
SSL Command-line Interface Commands 490
Troubleshooting of Encrypted SSL Connections 491
Decoding SSL Traffic with Wireshark 492
Network Traffic Capture using NSTRACE.SH 493
Network Traffic Capture Using NSTCPDUMP 494
TCPDUMP Options 494
NSTRACE Syntax and Options 495
Troubleshooting with Filter Expressions 496
SSL Offload Troubleshooting 497
Access to SSL VIP Address Failing 497
Intermediate Certificate Not Linked 498
Troubleshooting Policies 499
Caching 500
Use of Telnet or Monitors to Check Ports 502
High Availability 503
Configuration Synchronization Failure 503
File Synchronization Failure 503
Unexpected Failover 504
Test Your Knowledge: Troubleshooting 506

© Copyright 2012 Citrix Systems, Inc.

Notices

Citrix Systems, Inc. (Citrix) makes no representations or warranties with respect to the content or use of this publication. Citrix specifically disclaims any expressed or implied warranties, merchantability, or fitness for any particular purpose. Citrix reserves the right to make any changes in specifications and other information contained in this publication without prior notice and without obligation to notify any person or entity of such revisions or changes.

© Copyright 2012 Citrix Systems, Inc. All Rights Reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the purchaser's personal use, without express written permission of:

Citrix Systems, Inc.

851 West Cypress Creek Road

Fort Lauderdale, FL 33309

USA

http://www.citrix.com

The following marks are service marks, trademarks or registered trademarks of their respective owners in the United States and other countries.

Mark	Owner
Adobe®, Flash®, Acrobat®	Adobe Systems Incorporated
Citrix®, Citrix Access Gateway™, Citrix Education™, EdgeSight®, NetScaler®, MyCitrix™, XenDesktop® , TriScale™, Xen™, XenCenter™, Cloud Gateway™, Citrix Application Firewall™, XenServer®	Citrix Systems, Inc.
DSA™	Digital Service Advisers, LLC
FreeBSD®	Free BSD Foundation
Google Chrome™	Google, Inc.
OpenView®	Hewlett-Packard Company
Intel	Intel Corporation
WhatsUp	Ipswitch, Inc.

Mark	Owner
Kerberos	Kerberos, LLC
Linux	Linus Torvalds
Active Directory®, Internet Explorer®, Microsoft®, SQL Server®, Windows®, Windows Server®, Excel®, PowerPoint®, Word®, Office®, MGSoft, Lync Server®, Exchange®, SharePoint®	Microsoft Corporation
Firefox®	Mozilla Corporation
UNIX®	The Open Group
OpenSSL®	The Open SSL Software Foundation, Inc.
Java®, JavaScript®, Oracle®	Oracle Corporation
Pearson VUE®	Pearson Education, Inc.
PCI®	PCI Security Standards Council, LLC
RSA™	RSA Data Security, Inc.
SAP™	SAP, Inc.
Secureauth®	Secureauth Corporation
Shibboleth®	University Corporation for Advanced Internet Development
SolarWinds™	SolarWinds Worldwide, LLC
Splunk™	Splunk, Inc.
SSH®	SSH Communications Security Corporation
Thawte™	Symantec Corporation
Toolwire®	Toolwire
VeriSign™	Verisign, Inc.
Wireshark™	Wireshark Foundation, Inc.

Other product and company names mentioned herein might be the service marks, trademarks or registered trademarks of their respective owners in the United States and other countries.

Credits

Instructional Designers:	Larry Barrios, Jeremy Boehl, Karen Bridgewater, Dustin Clark, Orlando Martinez, Christopher Rudolph
Product Specialists:	Brian Bustin, Andrew Garfield, George Komoto
Graphic Artist:	Joshua Jack
Manager:	Erin Smith
Editors:	Ben Goodman, Kathryn Morris
Translation Coordinator:	Yashica Burgess
Subject Matter Experts:	Gregg Anderson, Simon Barnes, Terry Chou, Colin Christy, Mahasweta Dey, Abhishek Gautam, Bino Gopal, Dave Gunn, David Jimenez, Henrik Johansson, Curtis Kegler, Henny Louwers, Archana Maheshwari, Sandeep Mehta, Mike Nelson, Ronan O'Brien, Senthil Periasamy, Craig Pickford, Marissa Schmidt, Muthukumar Shunmugiah, Mark Simmons, Erin Smith, John Smith, Jessy Strebel, Richard Todd, Lena Yarovaya, Sharin Yeoh, Tony Zhang

Module 1

Getting Started

© Copyright 2012 Citrix Systems, Inc.

Overview

The Citrix NetScaler product line optimizes delivery of applications over the Internet and private networks, combining application-level security, optimization, and traffic management into a single, integrated appliance. The NetScaler features that you enable and the policies you set are then applied to incoming and outgoing traffic.

After completing this module, you will be able to:

- Identify the capabilities and functionality of the NetScaler.
- Identify basic NetScaler architecture.
- Identify basic NetScaler hardware and components.
- Identify deployment considerations and decide which deployment configurations are best suited for specific scenarios.
- Perform the initial NetScaler setup and configuration.
- Obtain, install, and manage NetScaler licenses.

© Copyright 2012 Citrix Systems, Inc.

NetScaler Overview

Citrix NetScaler is an application switch that performs application-specific traffic analysis to intelligently distribute, optimize, and secure layer-4 through layer-7 (L4-L7) network traffic for web applications. For example, a NetScaler system makes load-balancing decisions on individual HTTP requests rather than on the basis of long-lived TCP connections, so that the failure or slowdown of a server is managed much more quickly and with less disruption to clients. NetScaler functionalities are broadly categorized into features, such as switching, security, protection, and farm optimization.

Switching

When deployed in front of application servers, a NetScaler system ensures ideal distribution of traffic. You can segment application traffic according to information in the body of an HTTP or TCP request, and on the basis of L4-L7 header information such as URL, application data type, or cookie. Numerous load-balancing algorithms and extensive server health checks improve application availability by ensuring that client requests are directed to the proper servers.

Security and Protection

NetScaler security and protection features protect web applications from application-layer attacks. A NetScaler system provides built-in defenses against denial-of-service (DoS) and distributed denial of service (DDoS) attacks and supports features that protect applications against legitimate surges in application traffic that would otherwise overwhelm the servers. An available, built-in firewall can protect web applications from application-layer attacks, including buffer overflow exploits, SQL injection attempts, and cross-site scripting attacks. In addition, the firewall provides identity theft protection by securing confidential corporate information and sensitive customer data.

Optimization

Optimization features offload resource-intensive operations such as Secure Sockets Layer (SSL) processing, data compression, client keep-alive, TCP buffering, and the caching of static and dynamic content from servers. Optimization improves server performance in the farm and therefore speeds up applications. A NetScaler system supports several transparent TCP optimizations, which mitigate problems caused by high latency and congested network links, accelerating the delivery of applications while requiring no configuration changes to clients or servers.

 © Copyright 2012 Citrix Systems, Inc.

Introduction to the NetScaler System

The NetScaler system is an integrated web application delivery controller that slashes server and bandwidth requirements, cutting the costs of delivering enterprise applications in half. The NetScaler system gives IT managers the ability to instantly tap unrealized efficiency gains across all phases of the application life cycle, without having to become application experts.

The NetScaler system functions as an application accelerator through caching and HTTP compression. It also provides advanced management using layer-4 through layer-7 load balancing and content switching functions. The NetScaler system also includes application security using a web application firewall, including PCI-DDS security mandated protections and SSL VPN. The NetScaler system offloads application and web servers to ensure application availability, increased security through SSL, and server consolidation. It reduces the total cost of ownership of web application delivery and optimizing the user experience.

© Copyright 2012 Citrix Systems, Inc.

NetScaler Functionality

NetScaler content switching and load balancing dramatically improve the throughput and scalability of an Internet application infrastructure by decoupling each application request/response flow from the underlying transport. Content switching and load balancing ensure the most efficient use of transport protocols and resources, even in a scenario where all the content is encrypted or compressed.

The NetScaler system manages the complete life cycle of the request/response transaction. With this management, the NetScaler system is uniquely equipped to direct and control application requests most efficiently, from the client to the server and back again.

Connection multiplexing (also known as connection reuse) allows the servers to handle much fewer connections than are received by the NetScaler system. The more efficient use of the HTTP specification provides a significant boost to the effective capacity of the server by reducing server CPU load. With this separation, the NetScaler system can use the TCP proxy architecture to multiplex and reuse the server-side TCP connection independently from a client-side connection. This reuse of established and idle server-side TCP connections reduces the TCP overhead on web servers.

 © Copyright 2012 Citrix Systems, Inc.

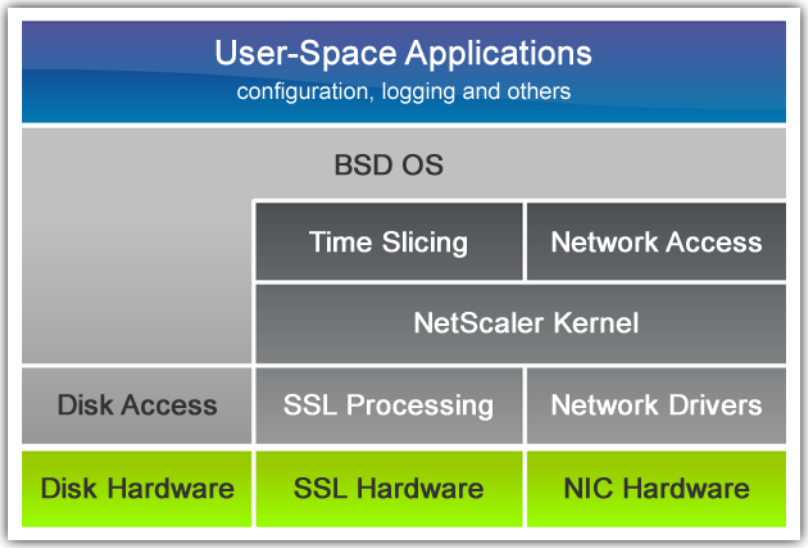
NetScaler Operating System Overview

The NetScaler operating system design is based on a layered module between the NetScaler kernel and the BSD operating system as shown in the figure. BSD and the NetScaler system share memory management. The NetScaler kernel operates below the BSD kernel and controls many of the features, including:

- Time slicing for BSD
- Network packet processing
- SNMP and syslog processing
- SSL offload

BSD manages several integral features of the NetScaler system, including:

- Startup process
- File system access
- Long-term logging
- Management processes



© Copyright 2012 Citrix Systems, Inc.

Hardware Platforms

NetScaler is available in three different appliance options to match the broadest range of business, performance, and deployment requirements.

- NetScaler MPX appliances are hardened network appliances that offer up to 50 Gbps performance.
- NetScaler SDX is a high-density consolidation platform that combines Xen-based virtualization and NetScaler's MPX architecture to run as many as 40 NetScaler instances simultaneously.
- NetScaler VPX virtual appliances run as virtual machines on a hypervisor, allowing NetScaler instances to be provisioned on demand.

For more information about NetScaler platforms, see Citrix eDocs at *http://edocs.citrix.com*

© Copyright 2012 Citrix Systems, Inc.

Hardware Compliance

NetScaler appliances are automatically compliant to the RoHS (Restriction of Hazard Substances) directives, as well as the WEEE (Waste Electrical & Electronic Equipment) directive. The NetScaler MPX appliance is SVHC (Substance of Very High Concern) compliant.

In addition, NetScaler MPX appliances are FIPS (Federal Information Processing Standard) compliant. For more information about FIPS-enabled NetScaler systems, see Citrix article CTX129543 at *http://support.citrix.com*.

Hardware Components

Each NetScaler system includes specific components. Some of these components are easily accessible on the exterior of a NetScaler appliance and should not be removed. The file systems are vital for the NetScaler system operations.

Network and Serial Interfaces

On all NetScaler platforms, the network interfaces are on the front of the appliance.

The RS232 serial console port on the front of each NetScaler appliance provides a direct connection between the appliance and a workstation or laptop, allowing direct access to the appliance for initial configuration or troubleshooting.

Redundant Power Supplies

All NetScaler MPX 7500 appliances and above have available redundant power supplies. These power supplies are interchangeable while the appliance is operating, which allows you to replace one power supply without shutting down the appliance, provided the other power supply is working.

File System

Important NetScaler data systems include:

- RAM Drive
- Flash Memory
- Hard Disk

The RAM drive is mounted on the root or `/file` system. During operation, binaries are executed from the RAM drive. The running configuration files for BSD level tools are used from the `/etc` file system residing on the RAM drive.

The flash memory is mounted in the NetScaler system as `/flash`. During startup, the RAM drive is initially populated from a compressed build file residing on the flash drive, and then the configuration files needed for BSD-level processes are copied from the `/nsconfig` directory on the flash drive into the `/etc` directory in the RAM drive. The `/nsconfig` directory is then linked to the `/flash/nsconfig` directory as a shortcut. When a save configuration command is executed, the running (in-memory) configuration is saved to this directory. The flash drive is on the back of the appliance; do not remove the flash drive.

The NetScaler system hard disk is used to store log data and core files and is used as long-term storage for unused builds. The `/var` directory represents the physical hard disk. The hard disk is on the back of the appliance; do not remove the hard disk.

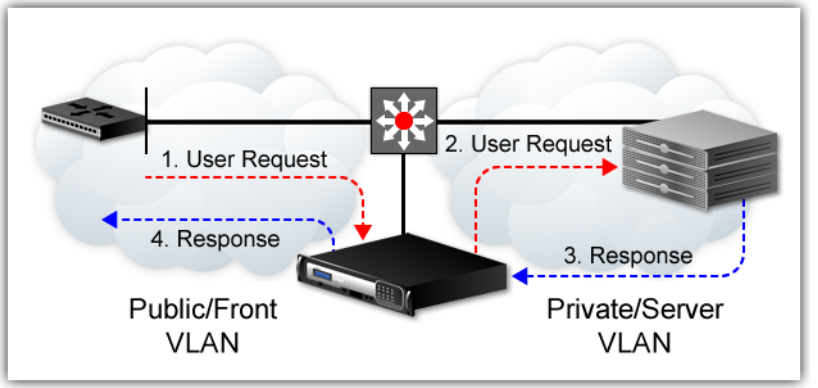
 © Copyright 2012 Citrix Systems, Inc.

Planning a NetScaler Deployment

You can deploy the NetScaler system in the following network topologies:

- One-arm mode
- Two-arm mode

One-Arm Mode



One-arm topologies are generally used to test the NetScaler system in a network. However, this topology requires most of the networking devices to be reconfigured to ensure that traffic passes through the NetScaler system. The two basic variations of one-arm topology are with a single subnet and with multiple subnets.

You can use a one-arm topology with a single subnet when the clients and servers reside on the same subnet. You can use a one-arm topology with multiple subnets when the clients and servers reside on the different subnets.

For example, consider a NetScaler system deployed in one-arm mode for managing three servers. The servers are connected to a switch on the network. A virtual server of type HTTP is configured on the NetScaler system, and HTTP services are running on the three servers. The three servers are on the private subnet, so a subnet IP address (SNIP) is configured through the NetScaler command line to communicate with them. The Use Subnet IP address (USNIP) option must be enabled so that the NetScaler system uses the SNIP instead of a MIP.

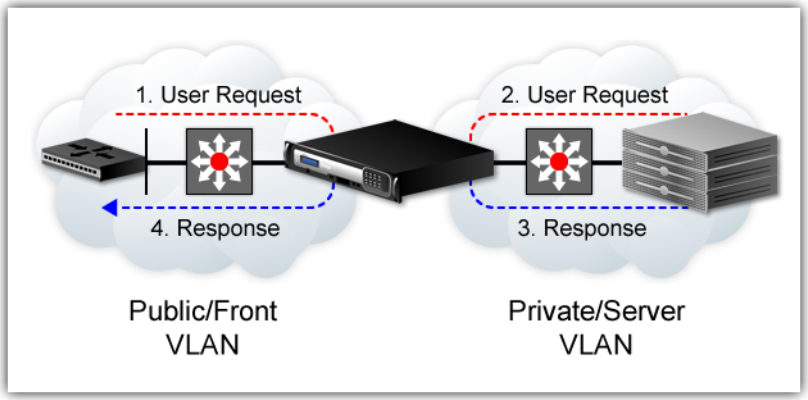
A one-arm mode configuration allows:

- A simple configuration with one physical interface and no risk of bridge loops.
- One or many VLANs with 802.1q tagging.
- Link aggregation to satisfy bandwidth requirements.

One-arm configurations are not recommended:

- When a server needs to see the real IP address of the client and the default gateway of the server cannot be changed to an IP address on the NetScaler system.
- When the servers are accessed directly and the default gateway has been changed to the NetScaler system, a situation that is sometimes addressed with the use of name-based services and explicit server definitions.

Two-Arm Mode



In a two-arm topology, one network interface is connected to the client network and another network interface is connected to the server network, ensuring that all traffic flows through the NetScaler system. The basic variations of two-arm topology are multiple subnets, typically with the NetScaler system on a public subnet and the servers on a private subnet, and transparent mode, with both the NetScaler system and the servers on the public network.

One of the most commonly used topologies has the NetScaler system inline between the clients and the servers, with a virtual server configured to handle the client requests. This configuration is used when the clients and servers reside on different subnets. In most cases, the clients and servers reside on public and private subnets, respectively. For example, consider a NetScaler deployed in two-arm mode for managing three servers, with a virtual server of type HTTP configured on the NetScaler system, and with HTTP services running on the servers. The servers are on a private subnet and a SNIP is configured on the NetScaler system to communicate with the servers. The Use Source IP (USIP) option must be enabled on the NetScaler system so that it uses the SNIP instead of the MIP.

Use transparent mode if the clients need to access the servers directly, with no intervening virtual server. The server IP addresses must be public because the client needs them. A NetScaler system is placed between the client and the server, so that the traffic must pass through the NetScaler system. You must enable layer-2 mode for bridging the packets. The NSIP and MIP are on the same public subnet.

Two-arm mode has the following benefits:

- It allows layer-3 (routed) deployments with split subnets.
- It allows layer-2 (bridged) deployments with one subnet on each side.

 © Copyright 2012 Citrix Systems, Inc.

- It supports transparent compression and SSL offload.
- It supports USIP address processing without server changes.

Two-arm topologies work in some situations in which a one-arm configuration does not work. With two-arm mode, you can use the NetScaler system to route some traffic as well as to balance load.

Test Your Knowledge

1. At which levels does the NetScaler system distribute, optimize, and secure network traffic?
    a. Layer 1 - layer 4
    b. Layer 2 - layer 5
    c. Layer 3 - layer 6
    d. Layer 4 - layer 7
2. Which NetScaler functionality allows servers in the environment to handle fewer connections than are received by the NetScaler system?
    a. Load balancing
    b. Content switching
    c. Connection multiplexing
    d. Global server load balancing
3. In which two situations would you want to set up the NetScaler system in two-arm mode? (Choose two.)
    a. You need to route traffic as well as to balance load.
    b. You need to use link aggregation to satisfy bandwidth requirements.
    c. You need to allow layer-2 deployments with one subnet on each side.
    d. You need to allow layer-4 deployments with split subnets.

© Copyright 2012 Citrix Systems, Inc.

Deployment Scenarios

You can integrate the NetScaler system into any network, either as a complement to existing load balancers, servers, caches, and firewalls, or as a standalone solution that can provide one or more of those functionalities. A successful NetScaler deployment requires planning for the correct deployment type, as well as a full migration of current network functions.

NetScaler deployment options include:

- Flex tenancy
- Multi-tenancy
- Displacement
- TriScale

Flex-Tenancy

You can use NetScaler physical and virtual appliances together in a two-tier, flex-tenancy architecture with each tier dedicated to managing specific actions.

The flex tenancy architecture segments web application delivery services into two tiers:

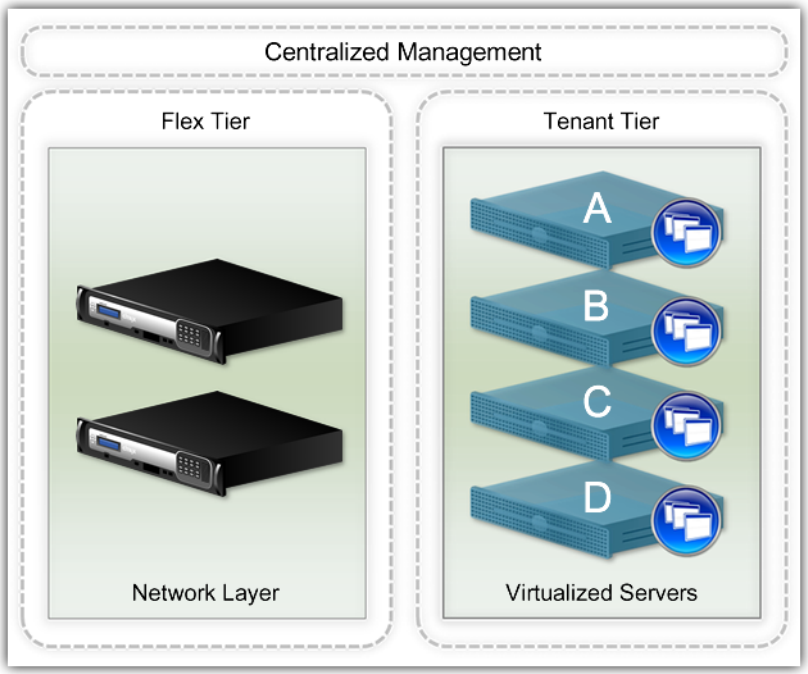
- Shared-network-service flex tier
- Application-specific tenant tier

The flex tier runs at the edge of the datacenter or an enterprise network. The flex tier performs application delivery services and associated policies that are common and applied to all applications. The tenant tier runs logically close to the applications, with each tenant getting its own NetScaler instance. The tenant tier isolates the application delivery needs that vary widely by application.

This architecture:

- Provides flexibility for how application delivery services are deployed.
- Makes it easier to shuffle applications and their associated application delivery resources on-demand without risking the stability of the entire network.
- Reduces costs.

© Copyright 2012 Citrix Systems, Inc.



Multi-Tenancy

With NetScaler SDX, each instance runs as a separate virtual machine with its own dedicated NetScaler kernel, CPU resources, memory resources, address space, and bandwidth allocation. Network I/O is done in a way that not only maintains total system performance but also enables complete segregation of each tenant's data-plane and management-plane traffic.

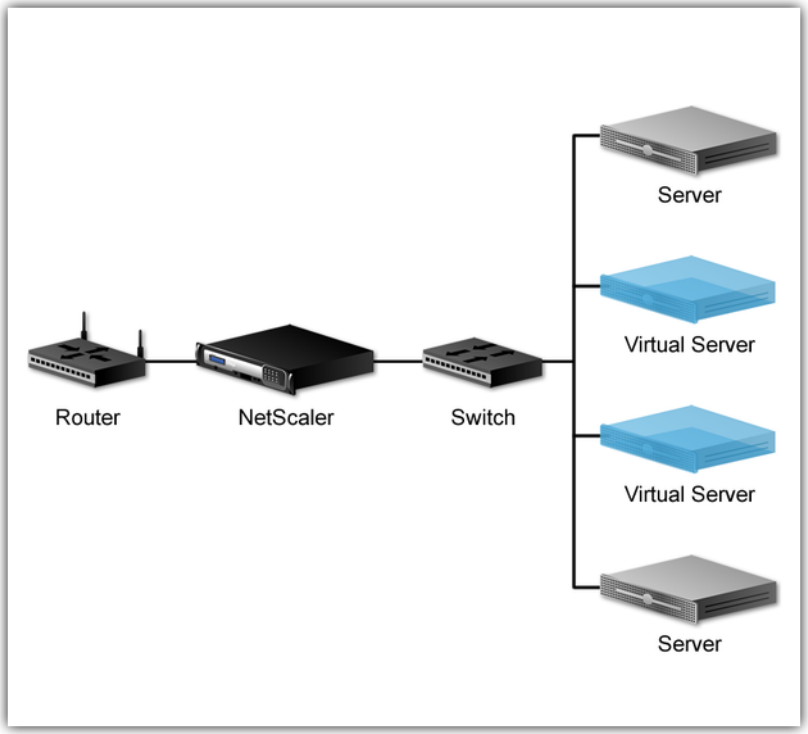
NetScaler SDX effectively delivers multiple virtual application delivery controllers (ADCs) by enabling fully isolated and independent NetScaler instances to run on a single appliance. A significant difference compared to other implementations of the virtual devices concept, however, is that NetScaler SDX provides complete isolation. As a result, memory, CPU cycles, and SSL cards are allocatable resources that you can move around and definitively assign to different NetScaler instances. This characteristic, namely complete isolation, is actually quite important given the application-centricity of the processing NetScaler performs.

Displacement

With displacement, a NetScaler system (inserted in-line) replaces another traffic manager and attempts to meet the configuration of the old device. The NetScaler system often provides superior performance and more compelling functionality compared to that offered by other traffic managers.

 © Copyright 2012 Citrix Systems, Inc.

You can deploy NetScaler systems in many scenarios to replace other products. As with a new deployment, it is critical with displacement to analyze and decide how to replicate the current functions of the network before deployment. Doing so ensures that network features are properly migrated to the new setup. You should explore new functionality only after you ensure the existing functions are replicated.



TriScale

Citrix TriScale technology uniquely incorporates multiple complementary methods for dynamically scaling application delivery infrastructure in response to changing conditions. Depending on the needs of an organization at any given time, you have the option to Scale Up, Scale Out, and Scale In.

Scale Up	With NetScaler Pay-As-You-Grow licensing, you can increase application delivery performance on-demand, up to the limit of the underlying appliance. You do not need to purchase additional hardware or pay for excess headroom in anticipation of future needs.

Scale Out	With TriScale Clustering, you can easily configure as many as 32 NetScaler systems to work in concert to deliver one or more applications. An implementation can start small yet scale beyond one Tbps in total capacity.
Scale In	With NetScaler SDX, as many as 40 fully isolated NetScaler instances can run on a single physical platform. You can scale the number of new tenants being supported while simultaneously simplifying the network by consolidating dedicated, often under-used application delivery controllers deployed for existing tenants.

© Copyright 2012 Citrix Systems, Inc.

NetScaler Configuration

A new NetScaler system is pre-configured with a default IP address (the NSIP) and associated subnet mask for management access. The default NSIP address is 192.168.100.1 and the subnet mask (netmask) is 255.255.255.0. You can change these values to fit the addressing scheme for your network. For your initial configuration, you must also specify at least one SNIP address or MIP address. Before saving your new configuration, you should change the administrator password. If you are setting up two NetScaler systems as a high availability pair, you configure one as primary and the other as secondary.

There are four ways to configure the NetScaler system:

- Using the LCD keypad
- Using the command-line interface
- Using the configuration utility
- Using the XML API

For more information about each of these configuration options, see Citrix article CTX132368 at *http://support.citrix.com*.

Logging in to the NetScaler System

The NetScaler administrator account is nsroot, and the default password is nsroot. The nsroot account is part of the default configuration. A best practice is to change the default password during installation.

You can access the NetScaler system using the GUI-based configuration utility or the command-line interface. You can configure the start menu and timeout session using the configuration utility logon window.

You can log on to the NetScaler system using the configuration utility when the system and software requirements for the workstation have been met.

You can access the command-line interface directly by connecting a workstation to the NetScaler serial port or by using SSH to connect to the NetScaler IP address.

 © Copyright 2012 Citrix Systems, Inc.

NetScaler Licenses

You must properly license a NetScaler system before you can deploy it to distribute, optimize, or secure network traffic for web applications. After you have obtained the licenses, you must install the licenses on your appliance, and then verify that you have enabled the features corresponding to the licenses.

There are four possible licenses:

- NetScaler platform license
- NetScaler features license
- Access Gateway Universal license
- Access Gateway ICA license

The NetScaler platform license is responsible for enabling all necessary features and as many as five SSL VPN connections. This license is allocated by default to hostname "ANY". The rest of the NetScaler licenses need to be allocated to the Host ID (MAC) of the appliance in order to enable the corresponding features. In the case of high availability, two licenses will be required. For more information about licensing your NetScaler, see Citrix article CTX121062 at *http://support.citrix.com*.

The Access Gateway Universal license will allow you to increase SSL VPN concurrent usage so that you are not restricted to five SSL VPN connections. This license floats across high availability pairs. You need to allocate the universal license to the NetScaler Licensing Hostname, which you can configure in `/nsconfig/rc.conf`.

> This is not necessarily the same hostname as the one created by `set ns hostname`, unless you have specified it. By default the hostname in `/nsconfig/rc.conf` is "ns".

To also increase ICA connections, you must upload an Access Gateway platform license, which will increase ICA connections to 10,000. This license also floats across high availability pairs. You must allocate this license to the NetScaler Licensing Hostname. For more information about configuring unlimited ICA connections, see Citrix article CTX125567 at *http://support.citrix.com*.

To Install the NetScaler License

To install the licenses using the NetScaler command-line interface:

1. Open an SSH connection to the NetScaler system by using an SSH client, such as PuTTY.
2. Log on to the NetScaler system by using the administrator credentials.
3. Switch to the shell prompt, create a license subdirectory in the nsconfig directory (if it does not exist) and copy the new license file(s) to this directory.
4. Restart the NetScaler system.

Using the configuration utility:

1. In a web browser, type the IP address of the NetScaler system, such as `http://192.168.100.1`.
2. In **User Name** and **Password**, type the administrator credentials.
3. Go to **Configuration** > **System** , and then click **Licenses**.
4. In the **Licenses** pane, click **Manage Licenses**. If the `/nsconfig/license` directory does not exist, you are prompted to create it.
5. In the **Manage Licenses** dialog box, click **Add**.
6. In the **Select License Files** dialog box, go to the location of the license files and select the file you want to upload.
7. After your file is uploaded to the license directory, click **OK**.
8. Restart the NetScaler system.

 © Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

1. Which NetScaler deployment option allows you to deliver multiple virtual application delivery controllers by enabling isolated and independent NetScaler instances on a single appliance?
    a. Flex-tenancy
    b. Multi-tenancy
    c. Displacement
    d. TriScale
2. How many fully isolated NetScaler instances can you run on a single physical NetScaler platform?
    a. 20
    b. 30
    c. 40
    d. 50
3. What is the maximum number of NetScaler systems you can configure to work together in a cluster?
    a. 8
    b. 16
    c. 32
    d. 64
4. Which NetScaler license is necessary for allowing more than five SSL VPN connections?
    a. Access Gateway Universal license
    b. NetScaler platform license
    c. Access Gateway ICA license
    d. NetScaler features license

© Copyright 2012 Citrix Systems, Inc.

© Copyright 2012 Citrix Systems, Inc.

Module 2

Basic Networking

© Copyright 2012 Citrix Systems, Inc.

Overview

A NetScaler is usually deployed in front of a farm and functions as a transparent TCP proxy between client and servers, without requiring any client-side configuration. To facilitate efficient and secure access to server resources, a NetScaler uses a set of IP addresses collectively known as NetScaler-owned IP addresses. To manage your network traffic, you assign NetScaler-owned IP addresses to virtual entities that become the building blocks of your configuration.

The NetScaler system is fundamentally a TCP proxy that separates the client connections from the server connections and manages separate connection tables for client and server connections. The NetScaler responds to client connections that are targeted at servers residing behind it, hiding the network topography and enforcing traffic security by acting as a single gateway that clients use to access the network.

After completing this module, you will be able to:

- Explain how the NetScaler fits in your network topology.
- Identify the IP address types that can be assigned to a NetScaler and the purpose of each address.
- Trace the path that a packet takes when it travels through a NetScaler.
- Determine if your LAN needs to be split into multiple virtual LANs (VLANs) and configure VLANs.
- Identify when Network Address Translation (NAT) is necessary and configure NAT on the NetScaler.
- Identify the benefits and drawbacks of using IPv6 and configure IPv6 on the NetScaler.
- Identify the routing methods best suited for an environment and configure the NetScaler to use method.

© Copyright 2012 Citrix Systems, Inc.

OSI Networking Model

Physical Layer (1)	The physical layer provides the hardware with means of sending and receiving informational bit streams over a communication channel. This layer defines connector and interface specifications and cable requirements.
Data Link Layer (2)	The data link layer provides a physical address for devices so that they can access a network to send and receive data. This layer furnishes transmission protocol knowledge and management.
The Network Layer (3)	The network layer determines how packets are routed or forwarded from one route to another. Routes are dependent upon static tables that are built within the network, however each path is calculated dynamically at the beginning of the conversation. This layer provides an end-to-end logical addressing system.
The Transport Layer (4)	The transport layer provides communication (typically TCP or UDP protocols) between devices through a network. This layer handles error-checking, network segmentation, and multiplexing.
The Session Layer (5)	The session layer permits users to establish sessions that can log on to a remote timesharing system or transfer data between devices. This layer provides a service to manage communication control, synchronization, and token management.
The Presentation Layer (6)	The presentation layer determines how applications represent incoming and outgoing data. This layer includes encryption, compression, graphics formatting, and content translation.
The Application Layer (7)	The application layer provides the end-user interface for a device connected to a network. This layer is not synonymous with the application itself.

© Copyright 2012 Citrix Systems, Inc.

NetScaler Architecture Overview

The NetScaler system is fundamentally a TCP (layer-4) proxy that separates the client connections from the server connections and manages separate connection tables for client and server connections. The NetScaler system can provide great traffic optimization as a gateway device by multiplexing client connections to web servers.

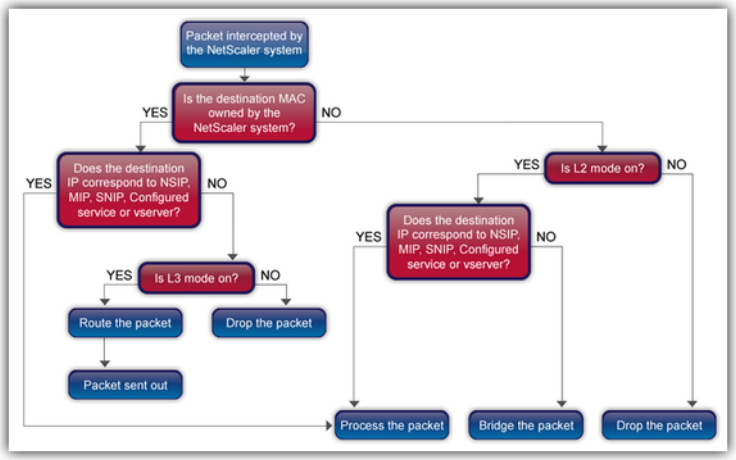
As a TCP proxy device, the NetScaler system responds to client connections that are targeted at servers residing behind it, hiding the network topography. The NetScaler system can enforce traffic security by acting as a single gateway that clients use to access the network.

The NetScaler system is not a UDP proxy. However, it still proxies the IP address, sourcing from a mapped IP address or subnet IP address as normal. This behavior can be turned on and off at a granular level.

© Copyright 2012 Citrix Systems, Inc.

Packet Forwarding

The NetScaler system can either route or bridge packets that are not destined for an IP address owned by the NetScaler (that is, the IP address is not the NSIP, a MIP, a SNIP, a configured service, or a configured virtual server). By default, layer-3 mode (routing) is enabled and layer-2 mode (bridging) is disabled, but you can change the configuration. The following flow chart shows how the NetScaler evaluates packets and either processes, routes, bridges, or drops them.



Layer-2 Mode

By default, the NetScaler system functions as a layer-3 network device. However, it can be configured to function as a layer-2 device. With layer-2 mode enabled, the NetScaler system functions as a bridge. Decisions are based on the MAC address of the packet (layer 2).

If the packet contains a destination MAC address that belongs to the NetScaler system, the NetScaler system will then look at the IP address destination to determine how to handle the traffic.

The layer-2 mode setting determines how the NetScaler system handles packets whose destination MAC address does not correspond to a NetScaler-owned MAC address. If layer-2 mode is disabled (default configuration), the NetScaler system drops any packet that is not destined for a NetScaler-owned MAC address.

If layer-2 mode is enabled, then the NetScaler system determines whether to process the packet or bridge the packet. The NetScaler system will process the packet if the destination IP address corresponds to a NSIP/MIP/SNIP address, configured service, or virtual server. Otherwise, the NetScaler system will bridge the packet, acting like a switch.

With layer-2 mode enabled, traffic that is not destined for the NetScaler system will be bridged to the networks that the NetScaler system is connected to. Note two exceptions to this behavior:

 © Copyright 2012 Citrix Systems, Inc.

- Broadcasts that are received on a NetScaler interface assigned to a VLAN are NOT forwarded to interfaces that are not assigned to a VLAN.
- ICMP and UDP traffic will be dropped if it exceeds the packet rate filters on the NetScaler system. The NetScaler system will not bridge the excess traffic requests.

Layer-3 Mode

The base behavior of the NetScaler system is to process the packet of any traffic sent to NetScaler-owned IP addresses. This means that any packet sent to an NSIP address, MIP address, SNIP address, configured service, or virtual server will be processed by the NetScaler system.

The layer-2 and layer-3 modes determine how the NetScaler system handles packets that are sent to an IP address that it does NOT own. These modes determine whether the NetScaler system should act as a switch and bridge the packets (layer-2 mode) and whether the NetScaler system should act as a router and forward the packets (layer-3 mode).

In the default configuration, layer-3 mode is enabled and layer-2 mode is disabled. These two modes are not exclusive.

MAC-based Forwarding Mode

MAC-based forwarding can be used to process traffic more efficiently by avoiding extraneous route/ARP lookups when forwarding packets. To avoid multiple lookups, the NetScaler system caches the source MAC address for every connection and returns the data to the same MAC address.

MAC-based forwarding is useful when using VPN devices because the NetScaler system ensures that the traffic flowing through the VPN passes through the same VPN device.

NetScaler-owned IP Addresses

The NetScaler system uses different types of IP addresses for management and proxying connections to the server. These IP addresses are:

- NetScaler IP (NSIP) addresses
- Subnet IP (SNIP) addresses
- Mapped IP (MIP) addresses
- Virtual IP (VIP) addresses

NetScaler IP Address

The NetScaler IP address (NSIP) is the primary address for management and general system access. The NetScaler IP address is a unique IP address. When NetScaler systems participate in a high-availability configuration, the NSIP address is used for primary communication between members of the high-availability configuration, and the NSIP is the only active IP address on the secondary member in a high-availability pair. The NSIP can be accessed from any enabled interface on the NetScaler system.

An NSIP address must be configured on a new NetScaler system. The default IP address and netmask is 192.168.100.1/16 (255.255.0.0).

Configuring an initial NSIP address or changing the NSIP address or subnet mask requires a restart of the NetScaler system. When configuring changes using the command-line interface, save the configuration first, change the NSIP address, and then restart the NetScaler system.

Any external authentication request the NetScaler performs—like using an external LDAP Server or Radius Server—uses the NSIP IP address.

Configuring a NetScaler IP Address

You can configure NSIP addresses through the Setup Wizard or through the command-line interface.

Configuration Utility

The NSIP can only be changed by running the Setup Wizard located in the **System** > **Setup Wizard** node.

 © Copyright 2012 Citrix Systems, Inc.

Command-line Interface

The command below will start the text-driven utility that will allow you to configure and save the NSIP address/subnet.

```
config ns
```

Authentication traffic, such as LDAP and RADIUS, also uses the NSIP address.

Subnet IP Address

The subnet IP (SNIP) address is used in connection management and server monitoring. A SNIP address provides the NetScaler system with an Address Resolution Protocol (ARP) presence in subnets to which the system may not be directly connected.

A NetScaler system should have a SNIP address configured for every directly connected subnet. When a SNIP is added to a NetScaler system, a static route entry is automatically added to the NetScaler system routing table; this route identifies the SNIP address as the default gateway on the NetScaler system for the corresponding subnet.

The Use Subnet IP (USNIP) mode can affect how the SNIP address is used by the NetScaler system to communicate with servers. USNIP mode is enabled by default. When USNIP mode is enabled, the SNIP address functions as a proxy IP and is used by the NetScaler system for NetScaler-system-to-server communication. In this mode, the server will see the SNIP address as the source IP address in packets received from the NetScaler system.

If USNIP mode is disabled, the SNIP address is not used to send traffic from the NetScaler system to the servers. Instead, a mapped IP address must be available. In most environments, USNIP mode is left enabled.

Individual SNIP addresses can be enabled to allow management access. When management access is enabled, connections to the NetScaler command-line interface over SSH and connections to the web-based configuration utility can be made using the SNIP address (as if it were an NSIP address). Using management-enabled SNIP addresses allows you to connect to the NetScaler system from a subnet other than the one where the NSIP is located. It also simplifies managing NetScaler systems in a high-availability configuration, since only the primary unit will respond to the SNIP. Management access is not enabled by default. Unlike the NSIP address (but like every other type of IP address), SNIP addresses are only active on the primary unit of a high-availability pair and show as passive on the secondary unit.

If multiple SNIP addresses are present on a subnet, the NetScaler will alternate between the SNIP addresses in round-robin manner when communicating with servers.

You can configure SNIP addresses by defining an IP address and a subnet mask.

Configuration Utility

Expand the **Network** > **IPs** node.

Command-line Interface

```
add ns ip <IP address> <Subnet Mask> -type SNIP -
mgmtAccess [Enabled | Disabled]
```

Because of the default behavior of the NetScaler system, IP addresses are not associated directly with interfaces (MAC addresses). As a result, all SNIP addresses are available on all interfaces by default, making it very easy to connect the NetScaler system to networks. This behavior can be modified by configuring VLANs on the NetScaler system and binding interfaces and SNIP or MIP addresses to a VLAN.

Mapped IP Address

A mapped IP (MIP) address is used for external connections from the NetScaler system. MIP addresses are used for connectivity in the absence of a SNIP address. For example, the MIP address is the proxy IP address of last resort. MIP addresses, like SNIP addresses, are used as the proxy address for NetScaler system-to-server communication. MIP addresses are still used even when the USNIP mode is globally disabled.

Citrix recommends to have at least one MIP address configured in the same subnet as the NSIP address. However, MIP addresses are not required.

The MIP address should be available across all subnets and should never be bound to a VLAN. It is only active on the primary unit of a high-availability pair - like every other IP address on the system other than the NSIP address - and shows as passive on the secondary unit.

When both a MIP address and a SNIP address are configured on the same subnet, the NetScaler system will use the SNIP address to communicate with servers by default (since USNIP mode is enabled). If USNIP mode is disabled, the MIP address will be used.

If multiple MIP addresses are present on a subnet, the NetScaler will use the MIP addresses in a round-robin fashion.

Citrix recommends to have either a MIP or SNIP address configured in the same subnet as the NSIP address. Certain features of the NetScaler will not function correctly, such as Proximity-based Monitors in GSLB or H/A synchronization.

You can configure MIP addresses by defining an IP address and a subnet mask.

 © Copyright 2012 Citrix Systems, Inc.

Configuration Utility

Expand the **Network** > **IPs** node.

Command-line Interface

```
add ns ip <IP address> <Subnet Mask> -
mgmtAccess [Enabled | Disabled]
```

Virtual IP Address

Virtual IP (VIP) addresses are used for client-to-NetScaler-system communication. Virtual IP addresses are assigned to virtual servers on the NetScaler system. VIP addresses are generally presented to the clients as a logical abstraction of a physical server behind the NetScaler system.

When the VIP address is a public IP address, it usually corresponds to the DNS entry for a domain. A VIP address is automatically created when a virtual server is added. A virtual server is identified as a unique IP/port combination. A virtual server could be configured using an IP address (VIP1) on port 80 (HTTP), and a second virtual server could be configured using the same IP address (VIP1) on port 3389 (RDP). A single VIP address can support a maximum number of 64,000 available TCP ports, corresponding to a maximum of 64,000 separate virtual servers (distinct ports) on a single VIP address.

Disabling or changing the status of a VIP address will affect all virtual servers using the VIP address.

DNS Virtual IP Addresses

A NetScaler DNS virtual server can be used to load balance and thus respond to incoming DNS requests for the domain of a customer. In addition, any DNS requests for which NetScaler is authoritative can be sent to this same VIP, but rather than being load balanced, the request is intercepted and responded to by the ADNS of the NetScaler.

To Remove a NetScaler-Owned IP Address

You can remove any IP address except the NSIP. The following table provides information about the processes you must follow in order to remove the various types of IP addresses. Before removing a VIP, remove the associated virtual server.

IP address type	Implications
Subnet IP address (SNIP)	If the IP address being removed is the last IP address in the subnet, the associated route is deleted from the route table. If the IP address being removed is the gateway in the corresponding route entry, the gateway for that subnet route is changed to another NetScaler-owned IP address.
Mapped IP address (MIP)	If a SNIP exists, you can remove the MIPs. The NetScaler uses NSIP and SNIPs to communicate with the servers when the MIP is removed. Therefore, you must also enable use SNIP (USNIP) mode.
Virtual Server IP address (VIP)	Before removing a VIP, you must first remove the virtual server associated with it. For more information about removing the virtual server, see Citrix article CTX132359 at *http://support.citrix.com*.
GSLB-Site-IP address	Before removing a GSLB-site-IP address, you must first remove the site associated with it. For more information about removing the site, see Citrix article CTX132359 at *http://support.citrix.com*.

 © Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

Match the following terms with the correct descriptions.

- Mapped IP (MIP) address
- NetScaler IP (NSIP) address
- Subnet IP (SNIP) address
- Virtual IP (VIP) address

Description	Term
	Used in connection management and server monitoring.
	Used for external connections from the NetScaler system.
	The primary address for management and general system access.
	Used for client-to-NetScaler-system communication.

© Copyright 2012 Citrix Systems, Inc.

NetScaler Network Interfaces

Network interfaces in the NetScaler system are numbered in <slot>/<port> notation. After configuring your interfaces, you should display the interfaces and their settings to verify the configuration. You can also display this information to troubleshoot a problem in the configuration. In order to manage the network interfaces, you can enable and disable any interfaces. You can reset an interface to renegotiate its settings. You can clear the accumulated statistics for an interface. To verify the configuration, you can display the interface settings. You can also display the statistics for an interface to evaluate its health.

In addition to configuring individual interfaces, you can logically group interfaces using VLANs to restrict data flow within a set of interfaces, and you can aggregate links into channels. In a high-availability setup, you can configure a virtual MAC (VMAC) address if necessary. If you use layer-2 mode, you can modify the aging of the bridge table.

Enable or Disable Network Interfaces

Configuration Utility

1. In the Navigation pane, expand the Network node, then the Interfaces node.
2. In the Interfaces pane, select the network interface that you want to enable or disable, and do one of the following:
   - Enable the network interface
   - Disable the network interface

Command-line Interface

At the NetScaler command prompt, type one of the following pairs of commands to enable or disable an interface and verify the setting:

1. `enable interface <interface_num>`
2. `show interface <interface_num>`

or

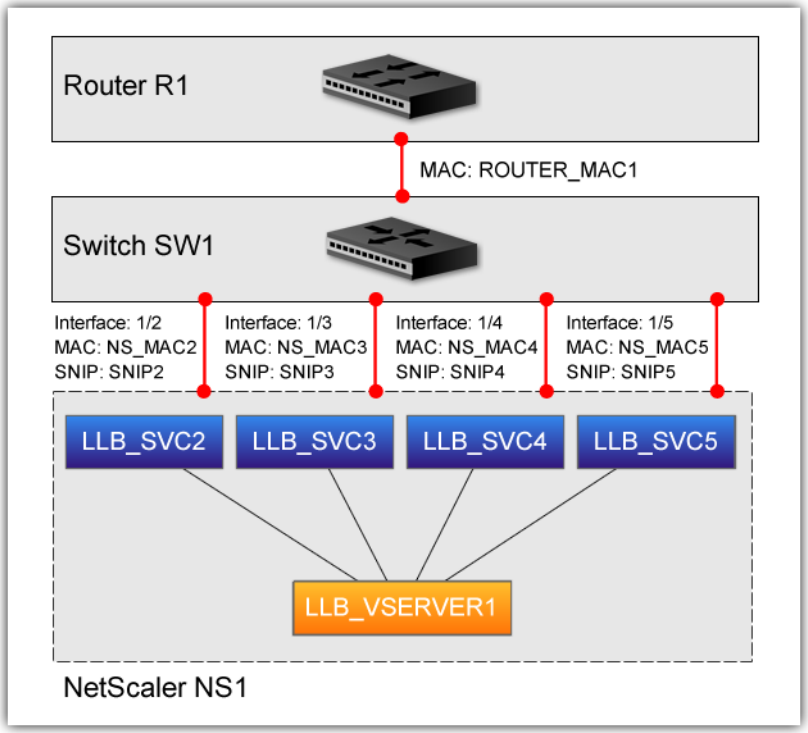
1. `disable interface <interface_num>`
2. `show interface<interface_num>`

SNIP Address Binding to Interfaces

You can bind a NetScaler-owned SNIP address to an interface without using layer-3 VLANs. With this configuration, any packets related to the SNIP address go only through the bound interface.

 © Copyright 2012 Citrix Systems, Inc.

This is useful in a scenario including a NetScaler system and an upstream layer-2 switch, where the upstream switch does not support Link Aggregation channels and you want the NetScaler system to load balance traffic originating from a server across the four links to the upstream switch.



MAC-based Forwarding Mode

MAC-based forwarding can be used to process traffic more efficiently by avoiding extraneous route/ARP lookups when forwarding packets. To avoid multiple lookups, the NetScaler system caches the source MAC address for every connection and returns the data to the same MAC address.

MAC-based forwarding is useful when using VPN devices because the NetScaler system ensures that the traffic flowing through the VPN passes through the same VPN device.

 © Copyright 2012 Citrix Systems, Inc.

NetScaler Connection Proxying

When a client initiates a connection, the NetScaler system terminates the client connection, initiates a connection to an appropriate server, and sends the packet to the server. The system does not perform this action for service type UDP or ANY.

You can configure the NetScaler to process the packet before initiating the connection with a server. The default behavior is to change the source and destination IP addresses of a packet before sending the packet to the server. You can configure the NetScaler to retain the source IP address of the packets by enabling Use Source IP mode.

© Copyright 2012 Citrix Systems, Inc.

Determining the Destination IP Address

Traffic sent to the NetScaler system can be sent to a virtual server or to a service. The system handles traffic to virtual servers and services differently. The NetScaler terminates traffic received at a virtual server IP (VIP) address and changes the destination IP address to the IP address of the server before forwarding the traffic to the server.

Packets destined for a service are sent directly to the appropriate server, and the NetScaler does not modify the destination IP addresses. In this case, the NetScaler functions as a proxy.

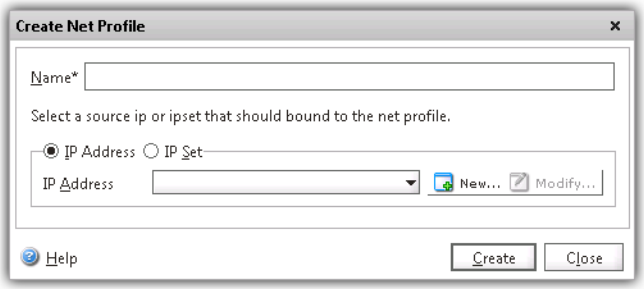
 © Copyright 2012 Citrix Systems, Inc.

Determining the Source IP Address

When the NetScaler system communicates with the physical servers or peer devices, by default it does not use the IP address of the client. NetScaler maintains a pool of mapped IP addresses (MIPs) and subnet IP addresses (SNIPs), and selects an IP address from this pool to use as the source IP address of a connection to the physical server. Depending on the subnet in which the physical server is placed, NetScaler decides whether a MIP or SNIP should be used.

If the Use Source IP (USIP) option is enabled, NetScaler uses the IP address of the client.

Specified Source IP Addresses for Back-end Communication Using Network Profiles



For communication with the physical servers or other peer devices, the NetScaler system uses an IP address owned by it as the source IP address. The NetScaler system maintains a pool of subnet IP addresses (SNIPs) and mapped IP addresses (MIPs) and dynamically selects an IP address while connecting with a server. Depending on the subnet where the physical server is placed, the NetScaler system decides whether a MIP or a SNIP should be used. This address pool is used for sending traffic as well as monitor probes.

In many situations, you may want the NetScaler system to use a specific IP address or any IP address from a specific set of IP addresses for back-end communications. The following are a few examples:

- A server can distinguish monitor probes from traffic if the source IP address used for monitor probes belongs to a specific set.
- To improve server security, a server may be configured to respond to requests from a specific set of IP addresses, or sometimes from a single specific IP address. In such a case, the NetScaler system can use only one of the IP addresses accepted by the server as the source IP address.
- The NetScaler system can manage its internal connections efficiently if it can distribute the MIPs or SNIPs into IP sets and use an address from a set only for connecting to a specific service.

© Copyright 2012 Citrix Systems, Inc.

To configure the NetScaler system to use a specified source IP address, create net profiles (network profiles) and configure the NetScaler entities to use the profile. A net profile can be bound to load-balancing or content-switching virtual servers, services, service groups, or monitors. A net profile has a SNIP or MIP address that can be used as the source IP address. It can be a single IP address or a set of IP addresses, referred to as an IP set. If a net profile has an IP set, the NetScaler system dynamically selects an IP address from the IP set at the time of connection. If a profile has a single IP address, the same IP address is used as the source IP.

Sending a Client IP Address to Servers

The NetScaler system usually functions in a transparent proxy configuration. Clients initiate connections to the NetScaler system using a VIP address. The NetScaler system terminates the connection from the client, processes the packet, and then initiates a connection to the appropriate server on behalf of the client.

The default behavior of the NetScaler system is to change the source and destination IP address of a packet received from a client before sending it to the server. The originating packet from the client contains the source IP address as the client IP address and the destination IP address as the virtual server IP address (the VIP on the NetScaler system). The NetScaler system then changes the packet before sending it on to the server so that the source IP address becomes the NetScaler MIP/SNIP address and a destination IP address becomes the physical IP address of the server.

As a result of proxying the connection, the server is unable to see the IP address of the client that originated the connection; the server can see only the NetScaler MIP or SNIP address. Since some applications require the client IP address for proper logging or functionality, the NetScaler system has two ways of providing this information to the server:

- Client-IP HTTP header insertion
- Use Source IP mode

 © Copyright 2012 Citrix Systems, Inc.

Use Source IP Mode

Use Source IP (USIP) mode is a networking mode in which the NetScaler system uses the actual client IP address to connect to the server and does not replace the source IP address in packets sent to the server with a MIP or SNIP address. The server will see the originating client IP address within the source IP field of the packet. While USIP mode will allow back-end resources to see and log the client IP addresses, there are several side effects and considerations when USIP mode is enabled.

By default, when proxying the connection the NetScaler system manages the connections to the server. For HTTP protocols, connection reuse and multiplexing result in improved efficiencies and performance. USIP mode severely limits multiplexing opportunities as each client will require its own connection to the server. Connection reuse, WAN latency optimizations, and denial-of-service (SYN) attack prevention are also limited.

© Copyright 2012 Citrix Systems, Inc.

Considerations and limitations when activating the Use Source IP mode feature include:

- USIP mode should not be enabled for NetScaler systems deployed in a one-arm configuration.
- Concurrent HTTP connection limits may be more of a factor since connections are not reused with USIP mode enabled. If HTTP connections will exceed 64,000 concurrent connections, USIP mode should be disabled.
- Multiplexing is only used for connections between the NetScaler and the server originating from the same source IP address, causing significantly more sessions to be established between the NetScaler system and the server. This behavior is inefficient for the NetScaler system and creates more overhead for the server.
- Surge protection is unable to function in a USIP mode environment, as the NetScaler system will be in a constant surge mode. The surge protection feature should be disabled if USIP is enabled.
- USIP mode requires routing in the environment to direct all of the server response traffic bound for the client IP address through the NetScaler system.

USIP mode does not eliminate connection reuse completely. The benefits of connection reuse are simply diminished. Connections from the same source IP address are reused. They are just a small fraction of the traffic the NetScaler system could be reusing.

If an application requires the original source IP address, then USIP mode can be enabled on a service-by-service basis.

 © Copyright 2012 Citrix Systems, Inc.

Configuring USIP Mode

USIP mode can be enabled or disabled globally (as a NetScaler mode) or it can be enabled or disabled for each service.

At the service level, the USIP setting is a global-override. Enabling or disabling the setting on an individual service overrides the global setting.

Whenever the global mode is changed, existing services are not modified; they retain their existing global-override setting. New services that are created will inherit the USIP setting from the global setting unless the value is explicitly overridden when configuring the service.

As a result, you may enable USIP mode broadly across all services or enable it specifically only for those services that require it.

> If an application requires a client IP address for logging or functional purposes, rely on client IP header insertion. If client IP header insertion is not suitable, enable USIP mode only on the specific service or services that require it. Keep the use of USIP mode as limited as possible.

Because of the limitations and considerations, do not enable USIP mode unless it is absolutely necessary.

Virtual Local Area Networks (VLAN)

A NetScaler system supports layer-2 port and IEEE 802.1q tagged VLANs. VLAN configurations are useful when you need to restrict traffic to certain groups of stations. You can configure a network interface as a part of multiple VLANs by using IEEE 802.1q tagging.

You can configure VLANs and bind them to IP subnets. The NetScaler then performs IP forwarding between these VLANs (if it is configured as the default router for the hosts on these subnets).

The NetScaler supports the following types of VLANs:

Port-based VLAN	The membership of a port-based VLAN is defined by a set of network interfaces that share a common, exclusive layer-2 broadcast domain. You can configure multiple port-based VLANs. By default, all network interfaces on the NetScaler are members of VLAN 1. If you apply 802.1q tagging to the port, the network interface belongs to a port-based VLAN. Layer-2 traffic is bridged within a port-based VLAN, and layer-2 broadcasts are sent to all members of the VLAN if layer-2 mode is enabled. When you add an untagged network interface as a member of a new VLAN, it is removed from its current VLAN.
Default VLAN	By default, the network interfaces on the NetScaler are included in a single, port-based VLAN as untagged network interfaces. This VLAN is the default VLAN. It has a VLAN ID (VID) of 1. This VLAN exists permanently. It cannot be deleted, and its VID cannot be changed. When you add a network interface to a different VLAN as an untagged member, the network interface is automatically removed from the default VLAN. If you unbind a network interface from its current port-based VLAN, it is added to the default VLAN again.

 © Copyright 2012 Citrix Systems, Inc.

Tagged VLAN

802.1q tagging (defined in the IEEE 802.1q standard) allows a networking device (such as the NetScaler) to add information to a frame at layer 2 in order to identify the VLAN membership of the frame. Tagging allows network environments to have VLANs that span multiple devices. A device that receives the packet reads the tag and recognizes the VLAN to which the frame belongs. Some network devices do not support receiving both tagged and untagged packets on the same network interface—in particular, Force10 switches. In such cases, you need to contact customer support for assistance. The network interface can be a tagged or untagged member of a VLAN. Each network interface is an untagged member of one VLAN only (its native VLAN). This network interface transmits the frames for the native VLAN as untagged frames. A network interface can be a part of more than one VLAN if the other VLANs are tagged. When you configure tagging, be sure to match the configuration of the VLAN on both ends of the link. The port to which the NetScaler connects must be on the same VLAN as the NetScaler network interface.

A Tagged VLAN configuration is neither synchronized nor propagated; you must perform the configuration on each unit in an HA pair independently.

VLAN Configuration

You can implement VLANs in the following environments:

- Single subnet
- Multiple subnets
- Single LAN
- VLANs (no tagging)
- VLANs (802.1q tagging)

If you configure VLANs that have only untagged network interfaces as their members, the total number of possible VLANs is limited to the number of network interfaces available in the NetScaler. If more IP subnets are required with a VLAN configuration, 802.1q tagging must be used.

When you bind a network interface to a VLAN, the network interface is removed from the default VLAN. If the network interfaces need to be a part of more than one VLAN, you can bind the network interfaces to the VLANs as tagged members.

You can configure the NetScaler to forward traffic between VLANs at layer 3. In this case, a VLAN is associated with a single IP subnet. The hosts in a VLAN that belong to a single subnet use the same subnet mask and one or more default gateways connected to that subnet. Configuring layer 3

for a VLAN is optional. Layer 3 is used for IP forwarding (inter-VLAN routing). Each VLAN has a unique IP address and subnet mask that together define an IP subnet for the VLAN. In an HA configuration, this IP address is shared with the other NetScaler systems. The NetScaler forwards packets between configured IP subnets (VLANs).

When you configure the NetScaler, you must not create overlapping IP subnets. Doing so impedes layer-3 functionality.

NSVLAN

NSVLAN is a VLAN to which the NetScaler management IP (NSIP) address's subnet is bound. The NSIP subnet is available only on interfaces that are associated with NSVLAN. By default, NSVLAN is VLAN1, but you can designate a different VLAN as NSVLAN. If you do so, you must restart the NetScaler system for the change to take effect. After the restart, NSIP subnet traffic is restricted to the new NSVLAN.

The traffic from the NetScaler IP subnet can be tagged (802.1q) with the VLAN ID specified for NSVLAN. You must configure the attached switch interface to tag and allow this same VLAN ID on the connected interface.

If you remove your NSVLAN configuration, the NSIP subnet is automatically bound to VLAN1, restoring the default NSVLAN.

Port-based VLANs

The membership of a port-based VLAN is defined by a set of network interfaces that share a common, exclusive layer-2 broadcast domain. You can configure multiple port-based VLANs. By default, all network interfaces on the NetScaler are members of VLAN 1.

If you apply 802.1q tagging to the port, the network interface belongs to a port-based VLAN. Layer-2 traffic is bridged within a port-based VLAN, and layer-2 broadcasts are sent to all members of the VLAN if layer-2 mode is enabled. When you add an untagged network interface as a member of a new VLAN, it is removed from its current VLAN.

Default VLAN

By default, the network interfaces on the NetScaler are included in a single, port-based VLAN as untagged network interfaces. This VLAN is the default VLAN. It has a VLAN ID (VID) of 1. This VLAN exists permanently. It cannot be deleted, and its VID cannot be changed.

When you add a network interface to a to a different VLAN as an untagged member, the network interface is automatically removed from the default VLAN. If you unbind a network interface from its current port-based VLAN, it is added to the default VLAN again.

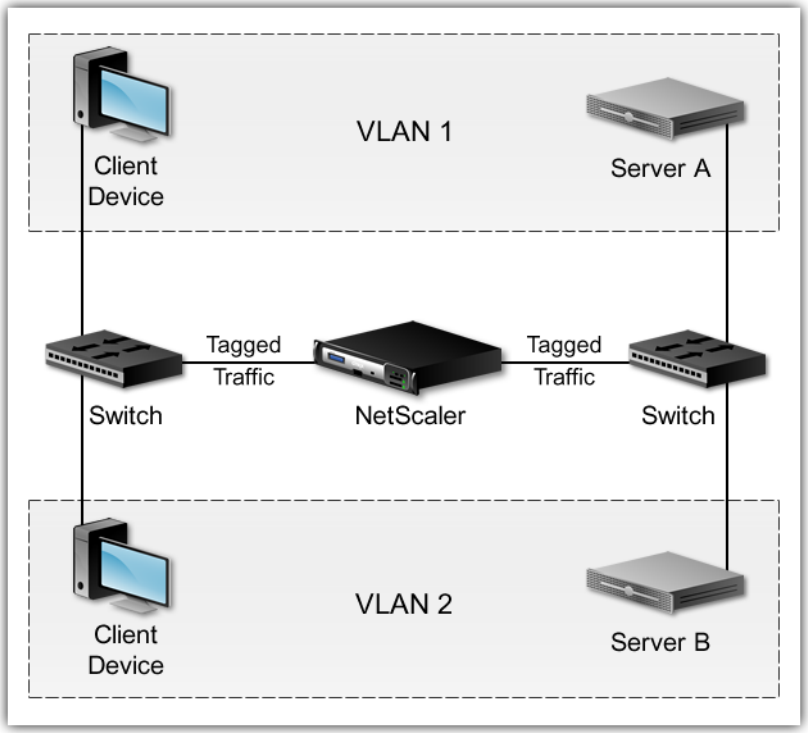
 © Copyright 2012 Citrix Systems, Inc.

Tagged VLANs

The figure shows a tagged VLAN structure. IEEE 802.1Q specifications define the tagging structure of VLAN traffic. VLAN tagging inserts an additional header between the layer-2 and layer-3 headers in the packet. The additional header contains a protocol ID and a VLAN ID. The virtual network with which the packet is associated is identified by the VLAN ID. This tagging information can be used by layer-2 VLAN-aware devices to intelligently forward the data to ports associated with the same network. The NetScaler system tagged VLAN functionality is fully 802.1Q compliant.

IEEE 802.1Q:

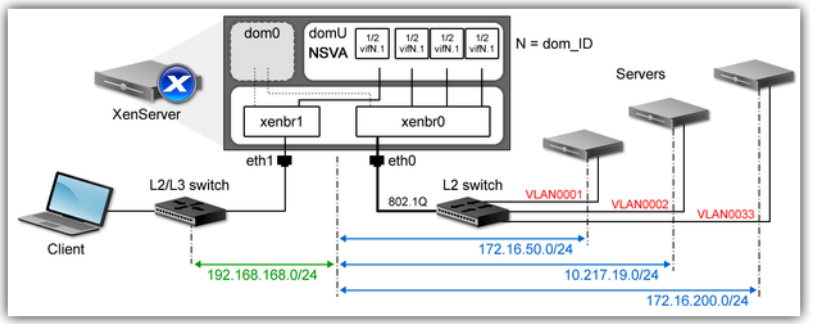
- Extra header between MAC header and IP header
- 2 byte protocol ID [TPID]
- 2 byte LAN ID [TCI]



Configuring VLANs

The figure shows further information for NetScaler VPX in a two-arm deployment.

You can configure a VLAN by defining a VLAN ID, an interface, and a SNIP.



To configure using the configuration utility:

- Expand the Network node and then the VLAN node.

To configure using the command-line interface:

```
show vlan id
```

```
bind vlan id[-ifnum InterfaceName
[-tagged]][-IPaddress IPAddress netmask]
```

 © Copyright 2012 Citrix Systems, Inc.

Network Address Translation (NAT)

Network address translation (NAT) involves modification of the source IP address, destination IP addresses or the TCP/UDP port numbers of IP packets that pass through the NetScaler system. It may involve the modification of any combination of all three.

Enabling NAT on the appliance enhances the security of your private network and protects it from a public network such as the Internet, by modifying source IP addresses of your network when data passes through the NetScaler. Also, with the help of NAT entries, your entire private network can be represented by a few shared public IP addresses. The NetScaler supports the following two types of network address translation:

- Inbound NAT (INAT), in which the NetScaler replaces the destination IP address in the packets generated by the client with the private IP address of the server.
- Reverse NAT (RNAT), in which the NetScaler replaces the source IP address in the packets generated by the servers with the public NAT IP addresses.

Inbound Network Address Translation

When a client sends a packet to a NetScaler system that is configured for inbound network address translation (INAT), the NetScaler system translates the public destination IP address of the packets to a private destination IP address and then forwards the packets to the server at that address.

When the server returns the packets, the NetScaler system translates the source IP address of the packets to the public source IP address and then sends the packets back to the client.

Coexistence of INAT and Virtual Servers

If both INAT and RNAT are configured, the INAT rule takes precedence over the RNAT rule. If RNAT is configured with a network address translation IP (NAT IP) address, the NAT IP address is selected as the source IP address for that RNAT client.

The default public destination IP in an INAT configuration is the virtual IP (VIP) address of the NetScaler system. Virtual servers also use VIPs. When both INAT and a virtual server use the same IP address, the virtual server configuration overrides the INAT configuration.

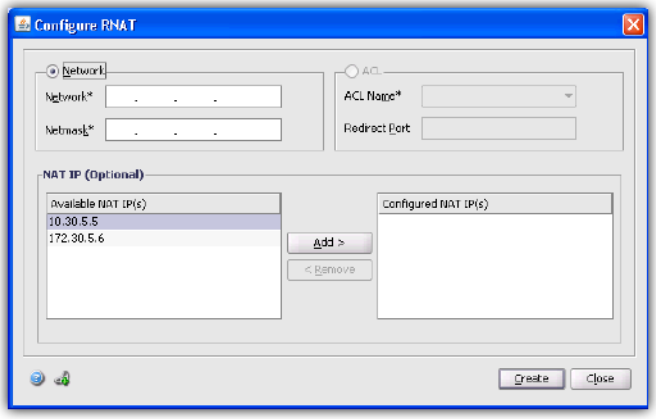
Following are a few sample configuration-setup scenarios and their effects.

© Copyright 2012 Citrix Systems, Inc.

Case	Result
You have configured a virtual server and a service to send all data packets received on a specific NetScaler port directly to the server. You have also configured INAT and enabled TCP. Configuring INAT in this manner sends all data packets received through a TCP engine before sending them to the server.	All packets received on the NetScaler system, except those received on the specified port, pass through the TCP engine.
You have configured a virtual server and a service to send all data packets of service type TCP that are received on a specific port on the NetScaler to the server after passing through the TCP engine. You have also configured INAT and disabled TCP. Configuring INAT in this manner sends the data packets received directly to the server.	Only packets received on the specified port pass through the TCP engine.
You have configured a virtual server and a service to send all data packets received to either of two servers. You are attempting to configure INAT to send all data packets received to a different server.	The INAT configuration is not allowed.
You have configured INAT to send all received data packets directly to a server. You are attempting to configure a virtual server and a service to send all data packets received to two different servers.	The virtual server configuration is not allowed.

 © Copyright 2012 Citrix Systems, Inc.

Reverse Network Address Translation (RNAT)



Reverse network address translation (RNAT) allows server-side addresses to be translated to the MIP address or a SNIP address of the NetScaler system when servers send data through the system. This behavior applies to connections that are initiated from the internal servers, as opposed to client connections passed through the NetScaler system.

RNAT does not alter the data portion of the communication in any way. As a result, if the application passes the host IP address as part of the data, that IP address is not the same as the address after the traffic has undergone RNAT. This incongruity causes that application to fail. For example, using the file transfer option in MSN Messenger is not possible through an RNAT session. The exception to this rule is File Transfer Protocol (FTP). Specific extended functionality has been put in place to support FTP through an RNAT session.

RNAT Port Re-use

The NetScaler RNAT is not only based on one port; it is also based on other socket data where the remote address and port are both taken into account. The NetScaler system can reuse the same port with the same IP address for multiple outgoing RNAT sessions.

© Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

1. When USIP mode is enabled for server communication, which address is used instead of a MIP or SNIP address?
    a. Client IP address
    b. Virtual IP address
    c. SNIP address
    d. Server IP address
2. Determine if the following statement is true or false. The NetScaler terminates packets destined for a service and changes the destination IP address to the IP address of the appropriate server before forwarding the packet to the server.
    a. True
    b. False
3. When the NetScaler system communicates with the physical servers or peer devices, by default it does not use the IP address of the client, but selects and IP address from which pool to use as the source IP address? (Choose two.)
    a. Mapped IP addresses
    b. Use Source IP addresses
    c. Subnet IP addresses
    d. Virtual IP addresses

 © Copyright 2012 Citrix Systems, Inc.

Module 3

Basic Load Balancing

© Copyright 2012 Citrix Systems, Inc.

Overview

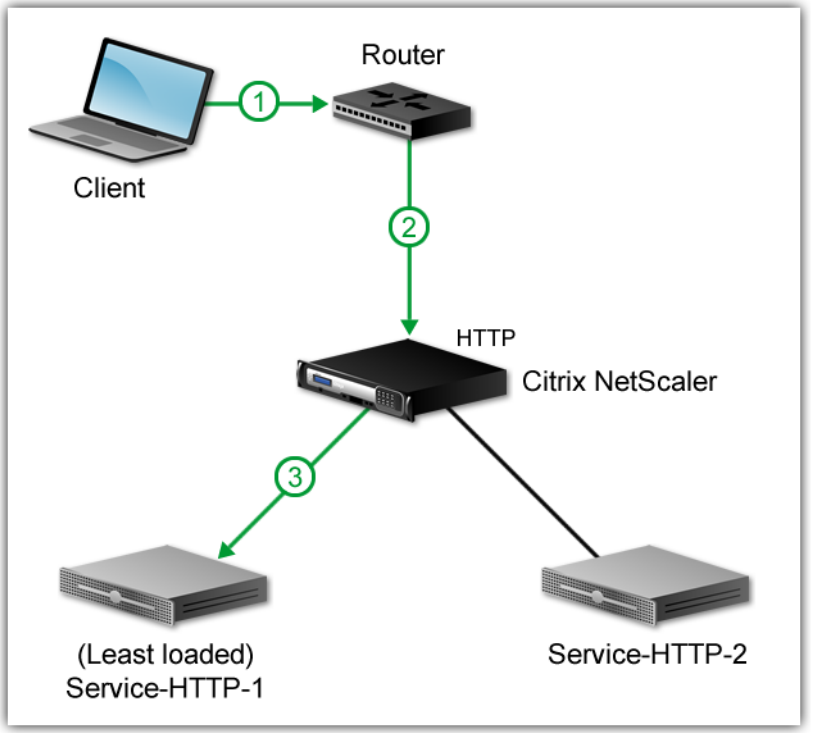
NetScaler load balancing distributes end-user requests for web pages and other protected applications across multiple servers that host or mirror the same content. You use load balancing primarily to manage end-user requests to heavily used applications, preventing poor performance and outages and ensuring that end users can access your protected applications. Load balancing also provides fault tolerance; when one server that hosts a protected application becomes unavailable, the feature distributes end-user requests to the other servers that host the same application.

After completing this module, you will be able to:

- Explain basic load-balancing concepts for a NetScaler system.
- Configure a basic load-balancing setup.
- Verify load-balancing configuration.
- Identify the different load-balancing methods.
- Identify the different session persistence methods.
- Explain why monitors are used in a load-balancing configuration.

© Copyright 2012 Citrix Systems, Inc.

Load-Balancing Basics



In a load-balancing configuration, the load-balancing virtual server is logically located between the client and the farm and manages traffic flow to the servers in the farm. On the NetScaler, the application servers are represented by virtual entities called services.

A load-balancing setup includes a load-balancing virtual server and multiple load-balanced application servers. The virtual server receives incoming client requests, uses the load-balancing algorithm to select an application server, and forwards the requests to the selected application server.

The load-balancing virtual server can use any of a number of algorithms, or methods, to determine how to distribute load among the load-balanced servers that it manages. The default load balancing method is the least connection method, in which the load-balancing NetScaler forwards each incoming client connection to whichever load balanced application server currently has the fewest active user connections.

The default LB method is least connection, but technically the NS uses round robin until enough connections are made to implement the least connection method.

 © Copyright 2012 Citrix Systems, Inc.

Entity Management

Entities in the NetScaler system are any configurable objects that are used with NetScaler features. Each entity is given a name when created. The entity name is used to associate the entity with a NetScaler feature. Entity names on the NetScaler system can have a maximum of 127 characters. The most commonly used entities are listed below.

Server	A server entity identifies a physical server and provides the IP address of the server. If you want to use the IP address of the server as the name of the server object, you can enter the IP address of the server when you create a service, and the server object is then created automatically. Alternatively, you can create the server object first and assign it an FQDN or other name, and then specify that name instead of the IP address when you create the service.
Service	A service entity can be a logical representation of the application server itself or of an application running on a server that hosts multiple applications. You define a service by an IP address, port, and protocol combination used to route requests to a specific load-balanced application server. The service identifies the type of traffic associated with a given server. You can configure multiple services for the same server. For example, you can configure a server to run HTTP, FTP, and TCP services/applications. The NetScaler system directs traffic to the server using the appropriate service. When you create a service, you associate it with a server. For load balancing, you bind services to virtual servers. Based on these services, the virtual servers will then load balance traffic across the available servers.
Load-Balancing Virtual Server	A virtual server is an aggregated system entity that usually comprises multiple servers and services. Rather than traffic being routed directly to the server, it is sent to a virtual server, which then makes a decision about which server to forward the traffic to based on the services bound to the virtual server. The state of the virtual server determines whether the client requests are accepted.

© Copyright 2012 Citrix Systems, Inc.

Monitor	A service monitor tracks a service and ensures that it is operating correctly. The monitor periodically performs a health check on each service that it is assigned. If the service does not respond within the time specified by the timeout, and a specified number of health checks fail, that service shows as DOWN. The NetScaler system then skips that service when performing load balancing until the issues that caused the service to quit responding are fixed.

Traffic Types

You can configure a load-balancing virtual server to support any number of traffic types. You can also configure multiple load-balancing servers on the same virtual IP address but for different services.

Application Protocols

HTTP, FTP, DNS, NNTP, and HTTPS are all application-layer protocols for which switching is performed at the request level rather than at the connection level. Request level switching allows greater granularity in the load-balancing process. The SSL setting represents HTTPS and instructs the virtual server to terminate and decrypt the HTTPS connection. You can configure a load-balancing server as type SSL and associate it with a service of type HTTP. The traffic from the end user is encrypted to the NetScaler system, and plain text HTTP traffic is forwarded from the NetScaler system to the server.

Session Protocols

You can associate the load-balancing server with a session protocol, either TCP or UDP. You can also secure TCP-based protocols other than HTTP using SSL. If the incoming traffic is SSL encrypted but not HTTP traffic, a virtual server of type SSL_TCP is created. This server decrypts the traffic on arrival and forwards it based on the protocols defined on the services bound to it.

General Traffic

You use ANY traffic type for any TCP, UDP, and Internet Control Message Protocol (ICMP) service. The configuration of ANY is used with some firewall load balancing and link load-balancing configurations where load balancing is time based.

The following table provides an overview of the load balancing traffic types.

 © Copyright 2012 Citrix Systems, Inc.

Application protocols	Session protocols	General traffic type
• HTTP • FTP • DNS • NNTP • SSL (HTTPS) • RTSP • SSL_BRIDGE	• TCP • UDP • SSL_TCP	ANY

Configure Basic Load Balancing

Before configuring your initial load balancing, enable the load-balancing feature. Then begin by creating at least one service for each server in the load-balancing group. With the services configured, you are ready to create a load-balancing virtual server and bind each service to the virtual server. That completes the initial setup. Before proceeding with further configuration, verify your configuration to make sure that each element was configured properly and is operating as expected.

Enable the Load-Balancing Feature

You can configure load balancing entities such as services and virtual servers when the load balancing feature is disabled, but they will not function until you enable the feature.

To Enable Load Balancing Using the Configuration Utility

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under Modes and Features, click **Change basic features**.
3. In the Configure Basic Features dialog box, select the Load Balancing check box, and then click **OK**.
4. In the Enable/Disable Feature(s)? message box, click **Yes**.

Load Balancing Enablement Using the Command-Line Interface

At the NetScaler command prompt, type the following command to enable load balancing and verify the configuration:

```
enable feature lb
```

```
show ns feature
```

 © Copyright 2012 Citrix Systems, Inc.

Server Creation

Before a NetScaler system can begin load balancing for physical servers in the environment, you must first create server and service entities that represent these servers.

You can create servers explicitly as named entities before creating services. You then create the service and reference the existing server object. However, if you create the service first, the system automatically creates a server object based on the IP address used when configuring the service.

When you create a server, you must identify it by specifying its IP address or domain name. If you specify the domain name, you can later change the IP address of the physical server without having to modify the entry in the list of servers on the NetScaler. The domain name is resolved to an IP address that is specified in an address record on the DNS.

For more information about creating a server, see Citrix article CTX132359 at *http://support.citrix.com*.

Maximum length is 255 characters.

Server Configuration Parameters

The following parameters will be used to configure a server.

name	The name assigned to the server. This alphanumeric string is required. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - period (.) colon (:) # and space ( ).
ipAddress	IP address of the server, in either IPv4 or IPv6 format. If the server is not reachable from the NetScaler system or is not active, the service shows as DOWN.
domain	Domain name that resolves to the IP address that represents the server.
ipv6Address	Resolve the domain name to an IPv6 address. Possible values: YES, NO. Default: NO.

state	The initial state of the server. Possible values: ENABLED, DISABLED. Default: ENABLED.
comment	A comment to help identify the server. Maximum length: 255 characters. To include spaces in a comment that you type on the NetScaler command line, enclose the entire comment inside quotation marks. The quotation marks become part of the comment. They are not required if you use the configuration utility.

To Create a Server Using the Configuration Utility

1. In the navigation pane, expand **Load Balancing**, and then click **Servers**.
2. In the details pane, click **Add**.
3. In the Create Server dialog box, specify values for the following parameters:
   - Server Name-name
   - IP Address-ipAddress
   - Domain Name-domain
   - Enable after Creating-state
   - Comment-comment
4. If you specify the domain name of the server and you want the domain name to be resolved to an IPv6 address, select the IPv6 Domain check box.
5. Click **Create** and then click **Close**.
   The server you named appears in the Servers pane.

Server Creation using the Command-Line Interface

At the NetScaler command prompt, type the following command:

```
add server name ipAddress | domain [-ipv6Address ( YES | NO )] -
state ( ENABLED | DISABLED )
 -comment string
```

Example of NetScaler Command Prompt to Create a Server

```
add server myserver 10.0.0.30 -state DISABLED -
comment "Web server for example"
```

 © Copyright 2012 Citrix Systems, Inc.

Services Configuration Overview

After you enable the load-balancing feature, you must create at least one service for each application server that is to be included in your load-balancing setup. The services that you configure provide the connections between the NetScaler and the load-balanced servers. Each service has a name and specifies an IP address, a port, and the type of data that is served. If you prefer to identify servers by name rather than IP address, you can create server objects and then specify the name of the server instead of its IP address when you create a service.

Services Creation

A service entity represents an application running on a server entity and is identified by a unique IP address/port combination. The service defines the traffic type between the NetScaler system and a server. Services must be bound to virtual servers before the NetScaler system is able to load balance incoming traffic between servers.

Before you create a service, you need to understand the different service types and how each is used. The following list describes the most common types of services supported on the NetScaler system. For more information about service types, see Citrix article CTX132359 on *http://support.citrix.com*.

ANY	Used for servers that accept any type of TCP, UDP, or ICMP traffic. The ANY parameter is used primarily with firewall load balancing and link load balancing.
DNS	Used for servers that accept DNS traffic, typically nameservers.
HTTP	Used for load-balanced servers that accept HTTP traffic, such as standard web sites and web applications.
FTP	Used for servers that accept FTP traffic. You can also use TCP or ANY service types for FTP servers.
SSL	Used for servers that accept HTTPS traffic, such as ecommerce web sites and shopping cart applications. The SSL service type enables the NetScaler system to encrypt and decrypt SSL traffic for your secure web applications.

© Copyright 2012 Citrix Systems, Inc.

TCP	Used for servers that accept many different types of TCP traffic, or that accept a type of TCP traffic for which a more specific type of service is not available. You can also use the ANY service type for these servers.
UDP	Used for servers that accept UDP traffic. You can also use the ANY service type.

When the service is created, the service type is specified by identifying the protocol/traffic type in use. The different service types available on the NetScaler system provide the ability to handle the traffic appropriately.

The use of both server entities and service entities allows the application running on a server to be abstracted from the physical entity. This abstraction allows multiple services running on a single server to be treated independently of each other.

Services are designated as DISABLED until the NetScaler system connects to the associated load-balanced server and verifies that it is operational. At that point, the service is designated as ENABLED. Thereafter, the NetScaler system periodically monitors the status of the servers and places any that fail to respond to monitoring probes, called health checks, back in the DISABLED state until they respond.

Service Configuration Parameters

The following parameters will be used to configure services.

name	Name of the service. This alphanumeric string is required and cannot be changed after the service is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - period (.) colon (:) # and space ( ).
serverName	Either the name of a previously created server object, or the IP address of the load-balanced server that hosts this service, in either IPv4 or IPv6 format. When you provide the IP address of the service, a server object is created with this IP address as its name. You can also create a server object manually and then select the server name instead of an IP address from the drop-down menu that is associated with this field. If the server is not reachable from the NetScaler or is not active, the service is designated as DOWN.

 © Copyright 2012 Citrix Systems, Inc.

serviceType	The type of connections that the service will handle. Possible values: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, and RTSP. Default: HTTP.
port	Port on which the service listens. The port number must be a positive number not greater than 65535.

To Create a Service Using the Configuration Utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, click **Add**.
3. In the Create Service dialog box, specify values for the following parameters:
   - Service Name-name
   - Server-serverName
   - Protocol-serviceType
   - Port-port
4. Click **Create** and then click **Close**.
   The service you created appears in the Services pane.

Service Creation Using the Command-Line Interface

At the NetScaler command prompt, type:

```
add service name serverName serviceType port
```

Example of NetScaler Command Prompt to Create a Service

```
add service Service-HTTP-1 192.0.2.5 HTTP 80
```

Virtual Server Creation

A virtual server provides the client with access to the actual servers behind the NetScaler system. The client connects to a virtual server which consists of an IP address, port, and protocol combination that accepts incoming traffic. The virtual server is bound to a number of physical services running on physical servers. These physical services consist of the IP address and port of the physical server. The client request is routed from the virtual server to the physical server based on the service that handles the inbound request. DNS is not required to make an arbitrary round-robin load-balancing decision; instead, DNS returns the VIP address of the load-balancing virtual servers to the client.

Multiple load-balancing virtual servers can be configured on the same VIP address as long as the combination of the VIP address, port, and service are unique.

The virtual server is designated as DOWN until you bind the services that you created to it and until the NetScaler system connects to those services and verifies that they are operational. Only then is the virtual server designated as UP.

Virtual Server Configuration Parameters

The following parameters will be used to configure virtual servers.

name	Name of the virtual server. This alphanumeric string is required and cannot be changed after the virtual server is created. The name must not exceed 127 characters, and the leading character must be a number or letter. The following characters are also allowed: @ _ - period (.) colon (:) # and space ( ).
IPAddress	IP address of the virtual server. This IP address can be an IPv4 or IPv6 address, and is usually a public IP address. Clients send connection requests to this IP address.
serviceType	The type of services to which the virtual server distributes requests. Possible values: HTTP, SSL, FTP, TCP, SSL_TCP, UDP, SSL_BRIDGE, NNTP, DNS, ANY, SIP-UDP, DNS-TCP, RDP, and RTSP. Default: HTTP.
port	Port on which the virtual server listens for client connections. The port number must be between 0 and 65535.

 © Copyright 2012 Citrix Systems, Inc.

To Create a Virtual Server Using the Configuration Utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the Create Virtual Server (Load Balancing) dialog box, specify values for the following parameters:
   - Name-name
   - IP Address-IPAddress
   - Protocol-serviceType
   - Port-port
4. Click **Create** and then click **Close**.
   The virtual server you created appears in the Load Balancing Virtual Servers pane.

Virtual Server Creation Using the Command-Line Interface

At the NetScaler command prompt, type:

```
add lb vserver name serviceType ip port
```

Example of NetScaler Command Prompt to Create a Virtual Server

```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
```

Services Bound to a Virtual Server

You can create most entities in the NetScaler system independently but they will not serve a function until they are bound to another related entity.

You can bind a service to a virtual server on the NetScaler system by specifying the virtual server and activating the service.

The state of the services bound to a virtual server determines the state of the virtual server: if all of the bound services are DOWN, the virtual server is marked DOWN, and if any of the bound services is UP or OUT OF SERVICE, the state of the virtual server is marked as UP.

To Bind a Service to a Load-Balancing Virtual Server by Using the Configuration Utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to bind the service.
3. Click **Open**.
4. In the Configure Virtual Server (Load Balancing) dialog box, click the **Services** tab, select the **Active** check box next to the service that you want to bind to the virtual server.
5. Click **OK**.

   You can bind a service to multiple virtual servers.

Services Bound to a Virtual Server Using the Command-Line Interface

At the NetScaler command prompt, type:

```
bind lb vserver name serviceName
```

Example of NetScaler Command Prompt to Bind a Service to a Load-Balancing Virtual Server

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
```

 © Copyright 2012 Citrix Systems, Inc.

Disabling Services

Services are enabled by default when you create them. You can disable or enable each service individually. When disabling a service, you normally specify a wait time, in seconds, during which the service continues to handle established connections, but rejects new ones, before shutting down. If you do not specify a wait time, the service shuts down immediately. During the wait time, the service's state is OUT OF SERVICE.

To Disable a Virtual Server Using the Configuration Utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server that you want to disable, and then click **Disable**.
3. In the Disable dialog box, click **Yes**.

Service Disablement using the Command-Line Interface

At the NetScaler command prompt, type:

```
disable service serviceName DelayInSeconds
```

Example of NetScaler Command Prompt to Disable a Service

```
disable service Service-HTTP-1 30
```

Removing Services

You can remove a service when it is no longer used. When you remove a service, it is unbound from its virtual server and deleted from the NetScaler configuration.

To Remove a Service Using the Configuration Utility

1. In the navigation pane, expand **Load Balancing**, and then click **Services**.
2. In the details pane, select the service that you want to remove, and then click **Remove**.
3. In the Remove dialog box, click **Yes**.

Removal of a Service using the Command-Line Interface

At the NetScaler command prompt, type:

```
rm service ServiceName
```

Example of NetScaler Command Prompt to Remove a Service

```
rm service Service-HTTP-1
```

 © Copyright 2012 Citrix Systems, Inc.

Configuration Verification

After finishing your basic configuration, you should view the properties of each service and load-balancing virtual server in your load-balancing setup to verify that each is configured correctly. After the configuration is up and running, you should view the statistics for each service and load-balancing virtual server to check for possible problems.

Viewing the Properties of a Server Object

You can view properties such as the name, state, and IP address of any server object in your NetScaler configuration.

- To view the properties of server objects by using the NetScaler command line, at the NetScaler command prompt, type:

  ```
  show server serverName
  ```

- To view the properties of server objects by using the configuration utility, in the navigation pane, expand **Load Balancing**, and then click **Servers**. The parameter values of the available servers appear in the details pane.

Viewing the Properties of a Virtual Server

You can view properties such as the name, state, effective state, IP address, port, protocol, method, and number of bound services for your virtual servers. If you have configured more than the basic load-balancing settings, you can view the persistence settings for your virtual servers, any policies that are bound to them, and any cache redirection and content-switching virtual servers that have been bound to the virtual servers.

- To view the properties of a load-balancing virtual server by using the NetScaler command line, at the NetScaler command prompt, type:

  ```
  show lb vserver name
  ```

- To view the properties of a load-balancing virtual server by using the configuration utility, in the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**. In the details pane, click a virtual server to display its properties at the bottom of the details pane.

Viewing the Properties of a Service

You can view the name, state, IP address, port, protocol, maximum client connection, maximum requests for each connection, and server type of the configured services; you can use this information to troubleshoot any mistake in the service configuration.

© Copyright 2012 Citrix Systems, Inc.

- To view the properties of services by using the NetScaler command line, the NetScaler command prompt, type:

```
show service name
```

- To view the properties of services by using the configuration utility, in the navigation pane, expand **Load Balancing**, and then click **Services**. The details of the available services appear on the Services pane.

Viewing the Bindings of a Service

You can view the list of virtual servers to which the service is bound. The binding information also provides the name, IP address, port, and state of the virtual servers to which the services are bound. You can use the binding information to troubleshoot any problem with binding the services to virtual servers.

- To view the bindings of a service by using the NetScaler command line, at the NetScaler command prompt, type:

```
show service bindings name
```

- To view the bindings of a service by using the configuration utility, in the navigation pane, expand **Load Balancing**, and then click **Services**. In the details pane, select the service whose binding information you want to view. Click **Show Bindings**. The bindings of the service you selected appear in the ServiceName dialog box.

 © Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

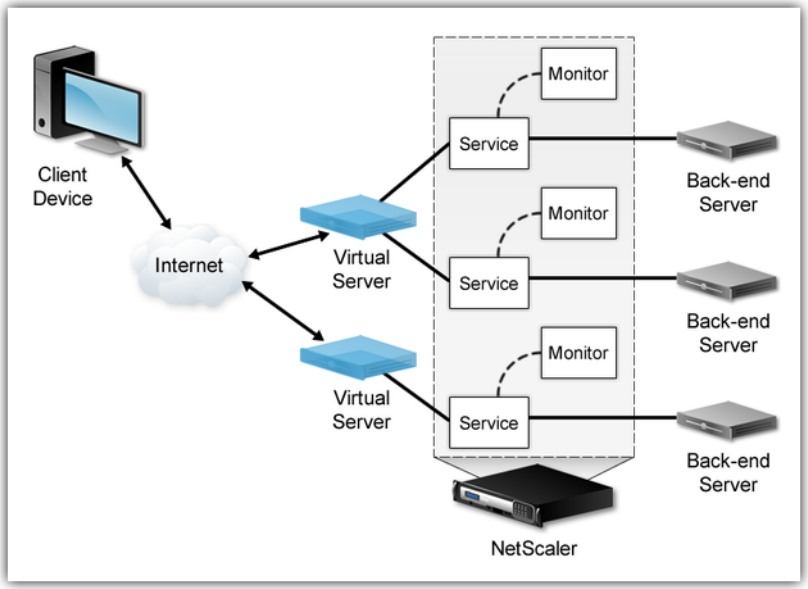
1. Which traffic types are associated with application protocols? (Choose three.)
    a. DNS
    b. HTTP
    c. SSL (HTTP)
    d. SSL_TCP
    e. RTSP
    f. TCP
    g. UDP
2. When creating a server, if you specify a domain name, you cannot change the IP address of the physical server unless you modify the entry in the list of servers on the NetScaler.
    a. True
    b. False
3. Multiple load-balancing virtual servers can be configured on the same VIP address as long as the following is true.
    a. All ports are the same.
    b. All services are the same.
    c. All VIP addresses, ports, and services are the same.
    d. The combination of VIP address, port, and service is unique.
4. The state of the services bound to a virtual server determines the state of the virtual server. Which of the states of the services will result in the state of the virtual server to be marked as UP?
    a. All of the bound services must be marked as UP for the state of the virtual server to be marked as UP.
    b. Any of the bound services are marked as UP for the state of the virtual server to be marked as UP.
    c. Any of the bound services are marked as OUT OF SERVICE for the state of the virtual server to be marked as UP.
    d. Any of the bound services are marked as UP or OUT OF SERVICE for the state of the virtual server to be marked as UP.
5. How does a service respond if you disable the service and specify a wait time?
    a. The service drops all connections, waits for the specified amount of time and then shuts down.
    b. The service continues to handle established connections, but rejects new ones during the specified amount of time and then shuts down.
    c. The service continues to handle existing and new connections for the specified amount of time and then shuts down.

© Copyright 2012 Citrix Systems, Inc.

d. The service will handle established connections until they have completed and will take new connections for the specified amount of time.

 © Copyright 2012 Citrix Systems, Inc.

Service Level Monitors



To manage a high-traffic load-balancing setup, the NetScaler system needs to track the state of services in near real time so that it can divert traffic from any load-balanced server that is not responding and send that traffic to a load-balanced server that is responding. Therefore, a monitor is bound to each service. The monitor is configured to test the service by sending periodic probes to the service. This is sometimes referred to as performing a health check. If the monitor receives a timely response to its probes, it marks the service as UP. If it does not receive a timely response to the designated number of probes, it marks the service as DOWN.

The load-balancing virtual server does not route requests to services that are DOWN. Those services are removed from its list of available services until they become available again and respond to monitor probes.

You can bind a single monitor or multiple monitors to the same service. If you bind multiple monitors to a service, they each evaluate responses to different types of traffic. The NetScaler system designates a service as UP only when the sum of the weights of all monitors bound to it and that are UP is equal to or greater than the threshold value configured on the service. The weight for a monitor specifies how much that monitor contributes to designating the service to which it is bound as UP.

The NetScaler system supports built-in monitors to monitor common types of services. It also supports end-user-created monitors based on the built-in monitors and allows you to create and configure custom monitors.

By default, monitors bound to services and service groups are enabled. When you enable a monitor, the monitor begins probing the services to which it is bound. If you disable a monitor bound to a

© Copyright 2012 Citrix Systems, Inc.

service, the state of the service is determined using the other monitors bound to the service. If the service is bound to only one monitor, and if you disable the monitor, the state of the service is determined using the default monitor.

Default Monitors

When you create a service group, the default monitor of the type appropriate for the group is automatically bound to it. Monitors periodically probe the servers in the service group to which they are bound and update the state of the service groups. They provide basic monitoring functionality and are limited to a single request/response.

You cannot delete or modify default monitors. When you bind any other monitor to a TCP service, the default monitor is unbound from the service.

The most commonly used default monitors are HTTP and Ping.

Built-in Monitors

The NetScaler system contains a number of built-in monitors that you can use to monitor your services. These built-in monitors handle most of the common protocols. You cannot modify or remove the built-in monitors; you can only bind a built-in monitor to a service and unbind it from the service.

For a complete list of built-in monitors and more information about them, see Citrix article CTX132359 at *http://support.citrix.com*.

Monitoring TCP-based Applications

The NetScaler system has two built-in monitors that monitor TCP-based applications: tcp-default and ping-default. The tcp-default monitor is bound to all TCP services; the ping-default monitor is bound to all non-TCP services.

Monitoring SSL Services

The NetScaler system has both a TCPS and HTTPS default secure monitor. You can use the secure monitors to monitor HTTP as well as non-HTTP traffic. The secure monitors work as described below:

- TCPS: The NetScaler system establishes a TCP connection. After the connection is established, the system performs an SSL handshake with the server. After the handshake is complete, the system closes the connection.
- HTTPS: The NetScaler system establishes a TCP connection. After the connection is established, the system performs an SSL handshake with the server. When the SSL connection is established, the system sends HTTP requests over the encrypted channel and checks the response codes.

 © Copyright 2012 Citrix Systems, Inc.

Monitoring FTP Services

To monitor FTP services, the NetScaler system opens two connections to the FTP server. It first connects to the control port, which is used to transfer commands between a client and an FTP server. After it receives the expected response, it connects to the data port, which is used to transfer files between a client and an FTP server. When the FTP server responds as expected on both connections, it is shown as UP.

The NetScaler system has two default monitors for FTP services: the FTP monitor and the FTP-EXTENDED monitor. The FTP monitor checks basic functionality; the FTPEXTENDED monitor also verifies that the FTP server is able to transmit a file correctly.

Monitoring SIP Services

The Session Initiation Protocol (SIP) is designed to initiate, manage, and terminate multimedia communications sessions. It has emerged as the standard for Internet telephony (VoIP). SIP messages can be transmitted over TCP or UDP. SIP messages are of two types: request messages and response messages.

Monitoring RADIUS Services

The NetScaler system RADIUS monitor periodically checks the state of the RADIUS service to which it is bound by sending an authentication request to the service. The RADIUS server authenticates the RADIUS monitor and sends a response. By default, the monitor expects to receive a response code of 2, the default Access-Accept response, from the RADIUS server. As long as the monitor receives the appropriate response, it marks the service UP.

Monitoring DNS and DNS-TCP Services

The NetScaler system has two default monitors that can be used to monitor DNS services: DNS and DNS-TCP. When bound to a service, either monitor periodically checks the state of that DNS service by sending a DNS query to it. The query resolves to an IPv4 or IPv6 address. That IP address is then checked against the list of test IP addresses that you configure. The list can contain up to five IP addresses. If the resolved IP address matches at least one IP address on the list, the DNS service is marked as UP. If the resolved IP address does not match any IP addresses on the list, the DNS service is marked as DOWN.

Monitoring LDAP Services

The NetScaler system has one default monitor that can be used to monitor LDAP services: the LDAP monitor. It periodically checks the LDAP service to which it is bound by authenticating and sending a search query to it. If the search is successful, the service is shown as UP. If the LDAP server does not locate the entry, a failure message is sent to the LDAP monitor and the service is shown as DOWN.

© Copyright 2012 Citrix Systems, Inc.

Custom Monitors

In addition to built-in monitors, you can create custom monitors, either based on the built-in monitors or from scratch, to check the state of your services. The NetScaler system provides several types of custom monitors based on scripts that are included with the NetScaler operating system that can be used to determine the state of services based on the load on the service or network traffic sent to the service. These are the inline monitors, end-user monitors, and load monitors.

With any of these types of monitors, you can use the supplied functionality, or you can create your own scripts and use those scripts to determine the state of the service to which the monitor is bound.

For more information about custom monitors, including inline monitors, end-user monitors, and load monitors, see Citrix article CTX132359 at *http://support.citrix.com*.

 © Copyright 2012 Citrix Systems, Inc.

Customization of Load-Balancing Configuration

After you configure a basic load-balancing setup, you can make a number of modifications to it so that it distributes load exactly as you need. You can modify the basic elements by changing the load-balancing algorithm, configuring load-balancing groups and using them to create your load-balancing configuration, configuring persistent client-server connections, configuring the redirection mode, and assigning different weights to different services that have different capacities.

For more information about customizing a load-balancing configuration, see Citrix article CTX132359 at*http://support.citrix.com*.

Load-Balancing Methods

Load-balancing decisions for incoming traffic are made based on the load-balancing method assigned to the service. The following sections provide available load-balancing methods. For more information about load-balancing methods, also called load-balancing algorithms, see Citrix article CTX132359 at *http://support.citrix.com*.

The following are common load-balancing methods:

Least Connections

Least connections is the default load-balancing method for the NetScaler system and is the most accurate method to distribute load intelligently. The least-connection calculation used to determine the service with the fewest number of connections is calculated differently depending on the service type. The following table provides an overview of the services used by the least-connections load-balancing method.

Service	Description
TCP, HTTP, HTTPS, SSL_TCP	Uses established, active connections to a service, and connections to a service waiting in the Surge Queue for the least connections calculation.
UDP	Uses all sessions between the client and the physical server for the least connections calculation. These sessions are logical, time-based entities and are created on the UDP packet that arrives first. If weights are configured, they are taken into account when server selection is completed.

© Copyright 2012 Citrix Systems, Inc.

Round Robin

Round robin is the simplest load-balancing method, distributing traffic based on a server-rotation system, regardless of load.

An incoming request is sent to Server 1, the next request is sent to Server 2, the next request is sent to Server 3, and when all servers have received a request, the cycle begins again.

This method is sufficient if all requests result in the same load on servers, but in most cases, a more robust load-balancing method based on metrics should be used.

Least Bandwidth

The least-bandwidth load-balancing method distributes traffic based on the amount of bandwidth being directed to each service. When this load-balancing method is configured, the NetScaler system routes traffic to the service currently serving the least amount of traffic. If weights are configured for a service, they are taken into account in the calculation to determine server selection.

Least Packets

The least-packets load-balancing method distributes traffic based on the rate of packet traffic to a service. The NetScaler system will make a load-balancing session based on the service currently serving the fewest packets for each second. If weights are configured for a service, then they are taken into account in the calculation to determine server selection.

Least Response Time

The least-response-time load-balancing method can be used with HTTP or HTTPS virtual servers to distribute traffic based on the average response time of each service. The response time, also referred to as Time to First Byte (TTFB), is the time interval between sending a request packet to a service and receiving the first response packet. A virtual server configured for least-response-time load balancing will direct traffic to the service with the shortest average response time. If weights are configured for a service, they are taken into account in the calculation to determine server selection.

The response-time calculation is always historical. As a result, it may not be a good metric for environments that have traffic that fluctuates greatly over short periods of time.

Hashing

The hashing load-balancing method refers to a number of different load-balancing methods that all use a hash value. The hash value is calculated based on a particular aspect of the client request, and the resulting value determines where to forward traffic. Hashing methods allow the virtual server to categorize requests and direct these requests types to the same physical resource. Depending on the hashing method used, the type of the requests can be controlled.

 © Copyright 2012 Citrix Systems, Inc.

Token

The token load-balancing method distributes traffic based on the value of a token extracted from a client request. For subsequent requests that contain the same token, the virtual server chooses the same physical server that handled the initial request.

Token-based load balancing is content-aware and operates differently depending on the type of service being used. For HTTP and HTTPS services, the token is found in the HTTP headers or URL and you configure it through an expression. For TCP services, tokens are extracted from the TCP payload.

The advantage of token load balancing is that it can be used across VIP addresses to ensure that requests within the same token are directed to the same physical servers, regardless of the protocol. The disadvantage is that it works only on the request. Token load balancing cannot tie the token to the server that the request was sent from, so token load balancing is generally not useful for jsessionid or other server-set tokens.

- This list of load-balancing methods covers only the basic methods.
- Server Application State Protocol (SASP) is supported for all load-balancing methods that use service weights.

Service Weights

In a load-balancing configuration, you assign weights to services to indicate the percentage of traffic that should be sent to each service. Services with higher weights can handle more requests; services with lower weights can handle fewer requests. Assigning weights to services allows the NetScaler system to determine how much traffic each load balanced server can handle, and therefore more effectively balance load.

Service weights can be configured for the following load-balancing methods:

- Least connections
- Round robin
- Least bandwidth
- Least packets
- Least response time

To Configure a Virtual Server to Assign Weights to Services Using the Configuration Utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server, and click **Open**.
3. On the **Services** tab, in the **Weights** spin box, type or select the weight to assign to the service.
4. Click **OK**.

Virtual Server Configuration to Assign Weights to Services Using the Command-Line Interface

At the NetScaler command line, type:

```
set lb vserver vServerName -weight Value ServiceName
```

Example of NetScaler Command Prompt to Configure a Virtual Server to Assign Weights to Services

```
set lb vserver Vserver-LB-1 -weight 10 Service-HTTP-1
```

 © Copyright 2012 Citrix Systems, Inc.

Persistence and Persistence Connections

Unless you configure persistence, a load-balancing stateless protocol, such as HTTP, disrupts the maintenance of state information about client connections. Different transmissions from the same client might be directed to different servers even though all of the transmissions are part of the same session. You must configure persistence on a load-balancing virtual server that handles certain types of web applications, such as shopping cart applications.

Before you can configure persistence, you need to know the difference between the different types of persistence, how they are used, and what the implications of each type is. You then need to configure the NetScaler system to provide persistent connections for those web sites and web applications that require them.

You can also configure backup persistence, which takes effect in the event that the primary type of persistence configured for a load-balancing virtual server fails. You can configure persistence groups so that a client transmission to any virtual server in a group can be directed to a server that has received previous transmissions from the same client.

Session Persistence Methods

Session persistence methods are determined based on the method you assign to the service. The following sections provide information about the most commonly used persistence methods.

Cookie-Insert Persistence

Cookie-insert persistence inserts an HTTP cookie in the client responses. The cookie is then used to direct subsequent requests from the client to the same physical server that handled the initial request. The cookie contains the IP address and port number of the service to which the client request is directed.

> If a client is unable to store a cookie, subsequent requests from that client will not have a cookie, and session persistence will not be honored. A load-balancing decision will be made based on the configured method.

The timeout value for cookie-insert persistence is the period of inactivity for a session. If you set the timeout value to zero, the session never expires in the NetScaler system; however, the expiration time is dependent on the client software used, and cookies usually expire when the client software is closed.

> If cookie-insert is configured as the primary persistence method and source-IP-address is configured as the backup, you can set different timeout values for each method.

Backup persistence is the method used when the primary persistence method fails. When you use cookie-insert persistence, you can configure source-IP-address persistence as the backup persistence

method to maintain persistence sessions. For example, if a client is using a web browser that does not support cookies, persistence is based on the source IP address of the client instead.

Source-IP-Address Persistence

Source-IP-address persistence routes a client request based on the configured load-balancing method, and then directs all subsequent requests from the same IP address to the physical server that handled the initial request.

> Traffic passing through a NAT device will appear to originate from the same IP address. If most client requests flow through a NAT device, source IP address persistence may not be the most efficient load-balancing method because every request will be directed to the same physical server even though the requests originate from multiple clients.

Source-IP-address persistence sessions are stored in a persistence table, which consumes system resources. A maximum of 256,000 entries can be stored in the persistence table at any one time.

The timeout value for source-IP-address persistence is the period of inactivity for a session. Once the timeout value is reached, the session is discarded, and server selection resumes based on the configured load-balancing method.

> If you configure cookie insert as the primary persistence method, and source-IP-address as the backup persistence method, you can set different timeout values for each method.

SSL Session ID Persistence

SSL-session-ID persistence is based on the arriving SSL session ID in a request, which is part of the SSL handshake process. Requests containing the same SSL session ID are directed to the physical server that processed the initial request.

SSL-session-ID persistence sessions are stored in a persistence table and therefore consume system resources. A maximum of 256,000 entries can be stored in the persistence table at any one time.

The timeout value for SSL-session-ID persistence is the period of inactivity for a session. Once the timeout value is reached, the session is discarded and server selection resumes based on the configured load-balancing method.

> SSL-session-ID persistence is not generally recommended as many web browsers renegotiate the SSL session as often as every two minutes. In such a case, the persistence is often broken. The SSL session is generally not long enough to use as a persistence criteria for typical web sessions.

 © Copyright 2012 Citrix Systems, Inc.

URL Passive Persistence

URL-passive persistence looks for an embedded server ID in client requests and then directs the request to the physical server that corresponds to the ID. Because this persistence method does not consume system resources, it can accommodate an unlimited number of persistent clients.

A server ID containing the IP address and port number of a physical server is embedded into the URL query of HTML links. When subsequent requests are made from the same client, the system extracts the server ID from the request and routes the traffic to the physical server that processed the initial request.

URL-passive persistence requires you to configure an expression that specifies the location of the server ID in the client request. URL-passive persistence does not use a timeout value. Persistence sessions are maintained as long as the server ID can be extracted from client requests.

Custom Server ID Persistence

Custom-server-ID persistence looks for an embedded server ID in client requests and then directs the requests to the physical server that corresponds to the ID. The server ID is an administratively configured integer, ranging from zero to 65,535, which is embedded into the URL query of HTML links. When a request is received that contains a server ID, the request is routed to the corresponding physical server.

Custom-server-ID persistence requires an that you configure an expression that specifies the location of the server ID in the client request.

If the server ID cannot be extracted from a request, a load-balancing decision is made based on the configured load-balancing method.

Custom-server-ID persistence does not use a timeout value. Persistence sessions are maintained as long as the server ID can be extracted from client requests.

Rule-based Persistence

Rule-based persistence evaluates a client request against a configured rule and creates a persistence session if the request is a match. If a request matches the contents of the rule, a persistence session is created, and all subsequent matching requests are directed to the physical server used to process the initial request. If the request does not match, it is routed to a server based on the configured load-balancing method.

Rule-based persistence requires you to configure an expression that defines the rule to be used in matching.

Rule-based persistence sessions are stored in a persistence table and, therefore, consume system resources. A maximum of 256,000 entries can be stored in the persistence table at any one time.

© Copyright 2012 Citrix Systems, Inc.

The timeout value for rule-based persistence is the period of inactivity for a session. Once the timeout value is reached, the session is discarded and server selection resumes based on the configured load-balancing method.

Destination IP Address Persistence

Destination-IP-address persistence is used with link load balancing. When you use this persistence method, the system selects a service based on the configured load-balancing method and then directs all subsequent packets that are destined to the same IP address as the first packet to the same service.

The timeout value for Destination-IP-address persistence is the period of inactivity for a session. Once the timeout value is reached, the session is discarded, and server selection resumes based on the configured load-balancing method.

Source/Destination IP Address Persistence

Source/destination-IP-address persistence is used with intrusion-detection-system (IDS) load balancing. When you use this persistence method, the system selects a service based on the configured load-balancing method and then directs all subsequent packets traveling from the same source to the same destination to the same physical service that received the first packet.

The timeout value for source/destination-IP-address persistence is the period of inactivity for a session. Once the timeout value is reached, the session is discarded and server selection resumes based on the configured load-balancing method.

SIP Call ID Persistence

With SIP Call ID persistence, the NetScaler system chooses a service based on the call ID in the SIP header. This enables it to direct packets for a particular SIP session to the same service and, therefore, to the same load balanced server. This persistence type is applicable specifically to SIP load balancing.

RTSP Session ID

With RTSP Session ID persistence, when the NetScaler system receives a request from a new client, it creates a new persistence session based on the Real-Time Streaming Protocol (RTSP) session ID in the RTSP packet header, and then directs the request to the RTSP service selected by the configured load-balancing method. It directs subsequent requests that contain the same session ID to the same service. This persistence type is applicable specifically to SIP load balancing.

 © Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

1. Which load-balancing method is most accurate at distributing load intelligently?
    a. least bandwidth
    b. least connections
    c. least packets
    d. round robin
2. Which of the load-balancing methods should not be used for environments that have traffic that fluctuates greatly over short periods of time?
    a. Least bandwidth
    b. Least packets
    c. Least response time
    d. Round robin
3. What does assigning service weights to services allow the NetScaler system to determine?
    a. The service with the fewest number of connections.
    b. The service currently serving the smallest amount of traffic.
    c. The percentage of traffic that should be sent to each service.
    d. The shortest average response time of a service.
4. Which persistence method is generally not recommended for typical web sessions?
    a. Custom Server ID Persistence
    b. Rule-based Persistence
    c. Source-IP-Address Persistence
    d. SSL Session ID Persistence
5. You cannot delete or modify default monitors.
    a. True
    b. False
6. Which of the following method is the default load-balancing method?
    a. least bandwidth
    b. least connections
    c. least packets
    d. round robin

© Copyright 2012 Citrix Systems, Inc.

© Copyright 2012 Citrix Systems, Inc.

Module 4

High Availability

© Copyright 2012 Citrix Systems, Inc.

Overview

A high availability (HA) deployment of two Citrix NetScalers can provide uninterrupted operation in any transaction. In a high-availability pair configuration, only one system is active. This system, which is known as the primary, actively accepts connections and manages servers. All shared IP addresses are active on the primary system only.

The secondary system monitors the health of the primary system. If the secondary system is in a healthy state, it is ready to actively accept connections if the primary system is experiencing issues. This process prevents downtime and ensures that the services provided by the NetScaler system remains available even if one system ceases to function.

The terms "primary" and "secondary" refer to the current state of a node, which can change as network conditions change. A way to specify a preference for primary status on either node, also known as preemption, does not exist.

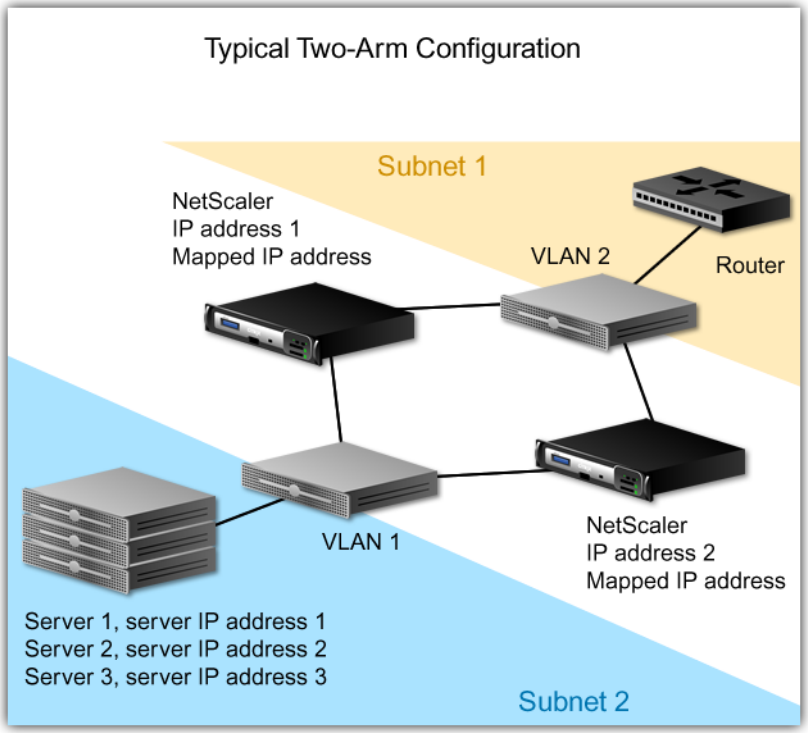
High availability packets are sent untagged by default, which can be an issue with a switch that handles tagged packets only.

After completing this module, you will be able to:

- Define high availability and the considerations for setup.
- Create a high-availability pair.
- Verify the high-availability setup.
- Configure high availability on the NetScaler system.

© Copyright 2012 Citrix Systems, Inc.

High Availability Functionality



The figure shows a typical two-arm high-availability configuration. Both systems in a high-availability pair exchange UDP heartbeat messages that communicate the state of the other and ensure that only one unit is taking traffic at a time. This configuration is known as active/passive. High availability does not require feature enablement; however, high availability needs to be configured.

If a component of the primary system fails, a message to the secondary system instructs it to take over as the primary system. If the secondary system does not receive a heartbeat message from the primary system after a specific interval of time, it automatically assumes the primary role. This scenario is known as failover.

> Three heartbeat messages must be missed for a failover to occur. High-availability heartbeat messages are sent every 200 milliseconds on UDP port 3003.

After a failover, all clients must reestablish their connections to the managed servers, but session persistence tables are synchronized between the primary and the secondary systems, honoring the previous persistence decision. For more information about TCP connection failover, refer to the Citrix NetScaler Networking Guide at *http://support.citrix.com*.

 © Copyright 2012 Citrix Systems, Inc.

Neither system is considered primary if the primary node experiences a failure and attempts to fail over to a secondary system that is not healthy. Therefore, it is possible for both systems to be secondary at the same time and for neither system to process traffic.

High Availability Node Configuration

A pair of NetScaler systems must be configured to become a high-availability pair. The process for configuring a high-availability pair involves first configuring the primary node then configuring the secondary node.

Citrix recommends that you set the status of the desired secondary node to stay secondary when nodes are configured. This practice ensures that an accidental failover does not occur during the configuration process, resulting in changes being made to the secondary rather than the primary node. Any changes that are made to the secondary node are not propagated to the primary node.

You should not use NetScaler VLANs when configuring high availability because it limits the NSIP address subnet. The NSIP address subnet should be available on all interfaces during high-availability configuration.

In a high-availability configuration, you can designate which interfaces to monitor for failing events. A failover occurs when any high-availability monitored interface goes down. If a particular interface is not being used, or if a failover is not required upon failure, the high-availability monitor should be disabled.

If an interface is not going to be used at all, a best practice is to disable it and turn off the associated high-availability monitor.

Pre-Configuration Checklist

Before configuring high-availability-pair nodes:

- Ensure that the NSIP addresses for the primary and the secondary nodes are unique from any other device on the network. The NSIP address can be changed using the set ns config command; this change requires a restart.
- Ensure that IP address conflicts can be viewed in the shell with the dmesg command.

While the RPC node password must be identical in a high-availability configuration, the nsroot password on both systems does not have to be the same.

Virtual Media Access Control Address

The primary and secondary nodes in a high-availability setup share the same virtual media access control (VMAC) address. The primary node owns floating IP addresses, such as the MIP, SNIP and VIP addresses, and responds to address resolution protocol (ARP) requests with its own MAC address. Therefore, the ARP table of an external device, such as an upstream router, is updated with the floating IP address and the MAC address of the primary node.

 © Copyright 2012 Citrix Systems, Inc.

The VMAC must be configured on both nodes of a high-availability pair; the nodes must have identical MAC addresses. When a failover occurs, the MAC address of the secondary node remains unchanged and the ARP tables on the external devices do not need to be updated.

When a failover occurs and the primary node is replaced by the secondary node, the replacement node will use the gratuitous address resolution protocol (GARP) to advertise the floating IP addresses that were learned from the original primary node. The MAC address that the new primary node advertises is the MAC address of its own network interface. Because some devices do not accept GARP messages, the external devices retain the IP address-to-MAC address mapping that the old primary node formerly advertised, which may result in a GSLB site becoming unavailable for those networks.

Configuring Primary and Secondary Nodes

You can use the following procedure to configure the primary and secondary nodes using the configuration utility or the command-line interface:

1. Disable the interfaces that are not connected or being used for traffic.
2. Disable monitoring for the interfaces whose failure should not cause a high-availability mode failover.
3. Assign a node ID to each NetScaler system.
4. Save the configuration.

   Before two nodes are able to function in a high-availability pair, they must first be made aware of each other.

   You join a new NetScaler system with an existing system to form a high-availability pair because Citrix recommends to enable "stay secondary" status on the new system. If "stay secondary" status is not enabled during the high-availability configuration, the new system may be elected to become the primary node, and the configuration settings on the existing system are replaced with the settings (typically a blank configuration) on the new system.

   When "stay secondary" status is enabled on the new system, the existing system becomes the primary node if it passes its health checks and its configuration settings are synchronized with the new system.

High-Availability Status

After you have configured two NetScaler systems in a high-availability pair, it is important that you verify the status of each node to ensure that the system is prepared in case of failover. For each node, the node state should show as UP and the master state should show as either primary or secondary, as appropriate. You should check that the sync state shows "success" on the secondary node because it verifies the successful completion of the configuration synchronization process.

Verifying Status on the Appliance

You can check the master status of a node remotely using either the configuration utility or the command-line interface. For example, the `show ha node` command will display the node state and master state. In addition, the LCD Configuration Display on the NetScaler appliance displays the status of the node.

 © Copyright 2012 Citrix Systems, Inc.

Propagation and Synchronization

Command propagation, enabled by default, causes any command issued on the primary node to execute on the secondary node. Consequently, you can test if high availability is working by making configuration changes to the primary node and then testing to see if the changes have been propagated to the secondary node.

Configuration synchronization occurs when a node comes up for the first time in a secondary state. The secondary node pulls the configuration from the primary node and overwrites its existing configuration.

In addition to automatic configuration synchronization, forced synchronization between two nodes is also supported. Forced synchronization is used whenever you want to ensure that changes made to a NetScaler configuration are transferred from the primary to the secondary system. For example, if a network interruption occurs and the systems are unable to communicate, a forced synchronization ensures that any changes made during the interruption are present on both systems.

In NetScaler releases 8.0 and later, if the nodes in a high-availability pair are running different versions of the NetScaler operating system, the node running the newest software version goes into listen mode. When a node is in listen mode, neither command propagation nor configuration synchronization occurs.

- Changes made using the `config ns` command in the command-line interface are not propagated. Any configurations made using this command must be performed on each node separately.
- High-availability configuration synchronization occurs on TCP port 3010.
- Command propagation between the primary and secondary occurs on TCP port 3011.
- The heartbeat messages are UDP packets sent to port 3003 of the other node in a high-availability pair.
- Secure high-availability configuration synchronization occurs on TCP port 3008.

Disabling Command Propagation

In some cases, command propagation may not be desired.

For example, if you make changes to the system configuration, and these changes cause problems, the changes are also made to the secondary node if command propagation is enabled.

When command propagation is disabled, you can first make changes on the primary node. Once the configuration is completed and is verified to be functioning properly, you can use the `force ha sync` command to ensure that the changes are propagated to the secondary node.

Automatic Configuration Synchronization

By default, configuration synchronization between the systems in a high-availability pair occurs automatically. Configuration synchronization occurs when:

- A node first comes up in the secondary state.
- A failover event occurs.
- A forced synchronization is issued.

When a save configuration is issued on the primary system, the running configuration on both systems is saved from memory to the "ns.conf" file on the flash drive. In some cases, automatic synchronization may not be desired.

For example, if you are testing two different configurations in a high-availability pair, the changes are also made to the secondary node if configuration synchronization is enabled. When configuration synchronization is disabled, you can first make changes on the primary node and then make changes on the secondary node. You can still fail over to the secondary node with the original configuration on the primary node.

Once the configuration is completed and is functioning properly, you can use the following command to make sure that the changes are synchronized in to the secondary node:

```
force ha sync
```

Forced Synchronization

Forced synchronization can be performed on either the primary or the secondary node; however, if synchronization is already in progress, the command fails and a warning message is displayed.

Before performing a forced synchronization, you must save the configuration because the saved configuration is copied over but the running configuration is not.

The saved configuration file is the "ns.conf" file. The running configuration is stored in memory and may contain unsaved configuration changes.

Forced synchronization will also fail when it is executed in the following circumstances:

- On a standalone NetScaler system
- On a NetScaler system on which high availability is disabled
- On a NetScaler system on which high-availability synchronization is disabled

The DONE message that is displayed after the `force ha sync` command is executed does not indicate that synchronization has been successful. To verify that synchronization is successful, execute the `show node` command to verify that the configurations on both nodes are the same.

 © Copyright 2012 Citrix Systems, Inc.

Performing a Forced Failover

When a failover occurs because the primary node fails, the primary node becomes secondary and the secondary node becomes primary. A forced failover is an administratively initiated failover on a node that produces the same results. A forced failover is desired if you want to test whether a high-availability pair configuration is performing correctly or to replace or upgrade the primary node.

Forced failover is performed in the configuration utility or with the `force ha failover` command in the command-line interface. The command can be issued on either the primary or the secondary node.

A forced failover will work only when:

- The primary node is able to determine that the status of the secondary node is UP.
- The health of the secondary node is good.
- The secondary node is not configured to stay secondary.

Performing a forced failover from the configuration utility may cause the client to lose connectivity with the NetScaler system. Citrix recommends that you perform a forced failover in the command-line interface.

High Availability Management

While a high-availability pair can be managed using the unique NSIP addresses assigned to each node during initial configuration, Citrix recommends that you manage the pair using either an MIP or SNIP address. When a shared IP address is used to connect to either the command-line interface or the configuration utility, the connection is made to the primary node in the pair. If a failover occurs, the secondary node becomes primary, and the same MIP or SNIP address then connects to the new primary node. Using an MIP or SNIP address is also helpful when you need to perform configurations from a subnet different from that of the high-availability pair.

Every NetScaler system is assigned an MIP address or a range of MIP addresses during initial configuration; however, management access must be enabled on the MIP or SNIP address before it can be used to manage a high-availability pair.

Citrix recommends that you secure management access in the NetScaler system.

Upgrading a High Availability Pair

When the nodes of a high-availability pair are running different versions of the system software, the node running the newest version goes into listen mode. When a node is in listen mode, neither propagation nor synchronization occurs.

Listen mode is used if you want to test the newest software version on one node before upgrading the second node. To conduct such a test, you perform a forced failover on the upgraded system, causing it to take over as the primary node. When a forced failover occurs, neither propagation nor synchronization occurs and all connections need to be re-established.

 © Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

1. Which task must you complete before performing a forced synchronization?
    a. Update the ns.conf file.
    b. Save the configuration.
    c. Copy the saved configuration.
    d. Copy the running configuration.
2. Which two IP address types have the option to enable management access? (Choose two.)
    a. NSIP address
    b. MIP address
    c. VIP address
    d. USIP address
    e. SNIP address
3. Which statement refers to a default high-availability setting?
    a. In a high-availability pair, management access and communication are secured through SSL.
    b. Synchronization between the systems in a high-availability pair has to be enabled after the nodes are connected.
    c. In a high-availability pair, both the primary and secondary NetScaler systems actively accept connections and manage servers.
    d. When the nodes of a high-availability pair are running different versions of the system software, the node running the newest version goes into listen mode.

© Copyright 2012 Citrix Systems, Inc.

© Copyright 2012 Citrix Systems, Inc.

Module 5

Policies and Expressions

© Copyright 2012 Citrix Systems, Inc.

Overview

For many NetScaler features, policies control how a feature evaluates data, which ultimately determines what the feature does with the data. A policy uses a logical expression, also called a rule, to evaluate requests, responses, or other data, and applies one or more actions determined by the outcome of the evaluation. Alternatively, a policy can apply a profile, which defines a complex action.

After completing this module, you will be able to:

- Implement compression policies to optimize traffic.
- Use templates to optimize web traffic from specific applications.
- Route traffic based on specific parameters.

© Copyright 2012 Citrix Systems, Inc.

Policies Overview

Some NetScaler features use default syntax policies, which provide greater capabilities than classic policies. You may have to manually migrate policies from classic to default (advanced) syntax if the classic policy features are now using advanced policies in the latest NetScaler release. This module describes the tool used to convert policies.

Policy Basics

Policies provide the foundation for the behavior of most NetScaler features, enabling the features to interpret the data, such as SSL requests, HTTP requests, TCP requests, and HTTP responses that pass through it. Typically, if a policy matches the data being evaluated, the NetScaler system takes one or more actions that are associated with the policy. For example, if an HTTP request matches a rewrite policy, the NetScaler can take an action to change (rewrite) the information in the request.

Most NetScaler features have built-in policies. The number and type of user-defined policies that a feature requires differs for each feature based on implementation requirements.

A NetScaler feature can use one of two policy types:

Classic policies	Classic policies evaluate basic characteristics of traffic and other data. For example, classic policies can identify whether an HTTP request or response contains a particular type of header or URL.
Advanced policies	Advanced policies can perform the same type of evaluations as classic policies. In addition, advanced policies enable you to analyze more data (for example, the body of an HTTP request) and to configure more operations in the policy rule (for example, transforming data in the body of a request into an HTTP header).

Basic Policy Components

Classic and advanced policy components include:

Name	Each policy has a unique name.

 © Copyright 2012 Citrix Systems, Inc.

Rule	The rule is a logical expression that enables the NetScaler feature to evaluate a piece of traffic or another object. For example, a rule can enable the NetScaler system to determine whether an HTTP request originated from a particular IP address, or whether a Cache-Control header in an HTTP request has the value No-Cache. Advanced policies can use all of the expressions that are available in a classic policy, with the exception of classic expressions for the SSL VPN client. In addition, advanced policies enable you to configure more complex expressions.
Bindings	Bindings ensure that the NetScaler system can invoke a policy when it is needed. You can bind the policy to one or more bind points.
Action	Policy evaluation ultimately results in the NetScaler system performing an action. For example, an integrated cache policy can identify HTTP requests for GIF or JPEG files. An action that you associate with this policy determines that the responses to these types of requests are served from the cache.

About Policy Bindings

A policy is bound to an entity that enables the policy to be invoked. For example, you can bind a policy to request time evaluation that applies to all virtual servers. A collection of policies with a common bind point constitutes a policy bank.

The following is an overview of different types of bind points for a policy:

Request time global	A policy can be available to all components in a feature at request time.
Response time global	A policy can be available to all components in a feature at response time.
Request time, virtual server-specific	A policy can be bound to request-time processing for a particular virtual server. For example, you can bind a request-time policy to a cache redirection virtual server to ensure that particular requests are forwarded to a load-balancing virtual server for the cache and other requests are sent to a load-balancing virtual server for the origin.

© Copyright 2012 Citrix Systems, Inc.

Response time, virtual server-specific	A policy can also be bound to response-time processing for a particular virtual server.
User-defined policy label	For advanced syntax policies, you can configure custom groupings of policies (policy banks) by defining a policy label and collecting a set of related policies under the policy label.
Other bind points	The availability of additional bind points depends on type of policy (classic or advanced syntax) and specifics of the relevant NetScaler feature. For example, classic policies that you configure for the Access Gateway have user and group bind points.

Policy Priorities

Lower-numbered policies have higher priority and are evaluated sooner. For example, if you have three policies with priorities of 10, 100, and 1000, the policy assigned a priority of 10 is performed first. All features except the rewrite feature on the NetScaler system implement only the first policy matching a connection. Setting the appropriate priority is key to a successful policy; the rewrite policy evaluates multiple policies in order of priority.

Citrix recommends that you leave room to add policies by setting priorities with intervals of 50 (or 100) between each policy.

Hypertext Transfer Protocol

Hypertext Transfer Protocol (HTTP) is the standard client-server protocol used to access web content. HTTP consists of a number of request methods that the client submits to the server and the response codes that the server returns along with any requested data.

By design, the HTTP protocol is stateless, meaning that from the point of view of the server, each request is fulfilled by a response on a one-to-one basis.

In web applications, requests may require incremental replies. For example, a request for some data may require the server to gather additional information before providing access to the resource. The initial request cannot be served with a single response. Additional requests and replies occur before the data is sent to the client. The server must be able to recognize that this series of requests and replies represents a single transaction rather than separate exchanges.

 © Copyright 2012 Citrix Systems, Inc.

Cookies

Web applications often use cookies to track the transaction state. Cookies are strings of data that the server sends to the client and they are included in client requests sent to the server. By examining cookie data, the server tracks the relationships between various requests made by the client.

SSL

HTTPS uses SSL security added at the Session Layer to encrypt data exchanged between client and server. HTTPS traffic is the same as HTTP traffic. The SSL encryption is wrapped around HTTP transactions. HTTP and HTTPS are identical in terms of requests and responses.

© Copyright 2012 Citrix Systems, Inc.

Expression Structures

An expression defines the types of requests and associated responses to which the NetScaler system applies an action. Expressions are either simple or compound.

Inline and named expressions are used in both classic and advanced policies. Inline expressions are part of a policy and cannot be reused by other policies. Named expressions are a common pool of logical statements applied to the content entering the NetScaler system. Named compound expressions are independent entities and can be reused by other policies.

> The NetScaler system uses a proprietary language based on Perl Compatible Regular Expression (PCRE) format that adds special terms, allowing aspects of a connection to be designated at a granular level.

Qualifiers, Operators, and Expression Values

Qualifiers specify what the policy examines. Operators determine how the qualifier will be evaluated. A qualifier is compared with the expression value, which can be literal text, a substring of text, or a numeric value.

Wildcards

The * wildcard character can be used to match a string within the specified qualifier. This character can appear only once within a string. You can restrict the processing of a string with wildcard characters. For example, the string `/*gif` will match on the first instance of gif, but not on further instances of gif if there is more than one in the string. It is important for strings to be matched properly for rule-based persistence.

Context-Sensitive Fields

The configuration utility displays context-sensitive fields depending upon the qualifier and operator selected.

If the HEADER qualifier is selected, type the appropriate header in the Header Name field. If the CONTENTS operator is selected, then type the length and offset in the Length and Offset fields.

The `length` and `offset` parameters are used to determine which part of a text string is examined for matching the operator and value. The `length` parameter controls the number of characters used and the `offset` parameter defines how far from the beginning of the field the capture begins. For example, if a string of “MSAccess” is examined by an expression with a length of 6 and an offset of 2, “Access” is the parsed string sent for evaluation.

 © Copyright 2012 Citrix Systems, Inc.

Simple Expressions

The simple expression is the basic building block of policies. A simple expression consists of a single logical comparison, such as SOURCEIP = 10.12.120.12. In this example, the qualifier is the source IP address of the traffic and the operator is “equal to.”

The following expression matches traffic with a source IP address of 65.219.20.0 and a netmask of 255.255.255.0:

```
add pol exp bad_boys “REQ.IP.SOURCEIP == 65.219.20.0 -
netmask 255.255.255.0”
```

The expression string is marked by quotation marks in the command-line interface and the ns.conf configuration file. The following expression matches requests for specific content in the URL:

```
add pol exp big_java “REQ.HTTP.URL contains big_job.jsp”
```

The following expression matches requests with URL lengths over 256 characters:

```
add pol exp big_url “REQ.HTTP.URLLEN > 256”
```

The following expressions match requests based on the HTTP request headers:

```
add pol exp accepts_html “REQ.HTTP.HEADER accept contains
text/html”
```

```
add pol exp has_cookie_header “REQ.HTTP.HEADER cookie exists”
```

Compound Expressions

Compound expressions can be made up of logical combinations of existing simple and compound expressions. Compound expressions use Boolean logic to compare the results of the separate expressions and determine if the compound expression evaluates to true or false.

The logical operators that can be used with compound expressions are:

- AND, represented by the double ampersand (&&)
- OR, represented by the double pipe (||)

In the NetScaler system, compound expressions are created from the same interface as simple expressions within the configuration utility.

Unlike simple expressions, compound expressions cannot be modified once they are created because of the risk of creating an expression loop.

The following compound expressions are built from simple expressions:

```
add pol exp big_bad "(bad_boys && big_java)"
```

```
add pol exp big_bad_or_long "(big_bad || big_url)"
```

Compound expressions cannot contain any qualifier other than COMPOUND. The COMPOUND qualifier does not need to be explicitly used. By referencing other expressions in the add expression statement, the statement assumes the use of COMPOUND== before the expressions. For example, the big_bad expression would be listed in the expression table as:

```
add policy expression big_bad COMPOUND== (bad_boys && big_java)
```

 © Copyright 2012 Citrix Systems, Inc.

Advanced Policy Conversion

You can convert a classic expression to the default expression syntax by using the nspepi conversion tool. You can also use the tool to convert all the classic expressions in the NetScaler configuration to the default syntax (with the exception of NetScaler entities that currently support only classic expressions).

The conversion tool does not convert policies configured for the following features, because the features currently support only classic policies:

- Authentication, Pre-authentication
- SSL
- Cache redirection
- VPN (session, traffic, and tunnel traffic)
- Content filtering (The Responder feature not only provides you with functionality that is equivalent to that provided by the content filtering feature but also surpasses the content filtering feature in the use cases that it supports. Additionally, Responder supports the more powerful default syntax for policy expressions.)

The following NetScaler features support both classic and default syntax expressions and, therefore, support the conversion of classic expressions to default syntax expressions:

- Application Firewall policies
- Authorization policies
- Named expressions
- Compression policies
- Content-switching policies
- User-defined, rule-based tokens/persistency (the `rule` parameter that is specified for a load-balancing virtual server)

For more information about how to convert features from classic to advanced policies, see Citrix article CTX131024 at *http://support.citrix.com*.

Test Your Knowledge

1. What are the four components of a policy? (Choose four.)
    a. Name
    b. Rule
    c. Label
    d. Bindings
    e. Priority
    f. Action
2. The NetScaler system has a list of policies and their priorities. Below is the list of assigned priorities for each policy. According to the assigned priorities, which policy will the NetScaler system perform first?
    a. 1050
    b. 150
    c. 550
    d. 50
3. Which aspect of an expression's structure specifies what the policy examines?
    a. Operator
    b. Expression value
    c. PCRE
    d. Qualifier
4. Which tool can you use to convert a classic expression to the default expression syntax?
    a. nspepi
    b. nsexpression
    c. nsprep
    d. nsconvert

 © Copyright 2012 Citrix Systems, Inc.

Module 6

Content Switching

© Copyright 2012 Citrix Systems, Inc.

Overview

Content switching allows the NetScaler system to direct traffic to servers based on the content that the user is attempting to access. Content switching involves configuring:

- Load-balancing servers
- Services
- Virtual servers
- Content-switching policies

At the end of this module, you will be able to:

- Describe content switching and the configuration process.
- Describe rule precedence in content switching.
- Configure policies and profiles.

© Copyright 2012 Citrix Systems, Inc.

Introduction to Content Switching

Content switching allows HTTP and HTTPS traffic requests to be intercepted and switched in a method that is transparent to the client. A NetScaler system can switch static and dynamic content.

Content switching provides the ability to direct traffic and client requests to back-end services based on an aspect of the request beyond the IP/port pair. Content switching allows the design of a complex internal system to appear to the public behind a single IP address. As clients connect to and request data from a single address, the NetScaler system examines the type of connection and sends it to the appropriate back-end service.

The NetScaler system diverts the application requests transparently to the client and the application, allowing the application to be managed separately from the hosting site.

When switching both static and dynamic requests, you must configure one load-balancing virtual server for static requests and a separate load-balancing virtual server for dynamic requests.

 © Copyright 2012 Citrix Systems, Inc.

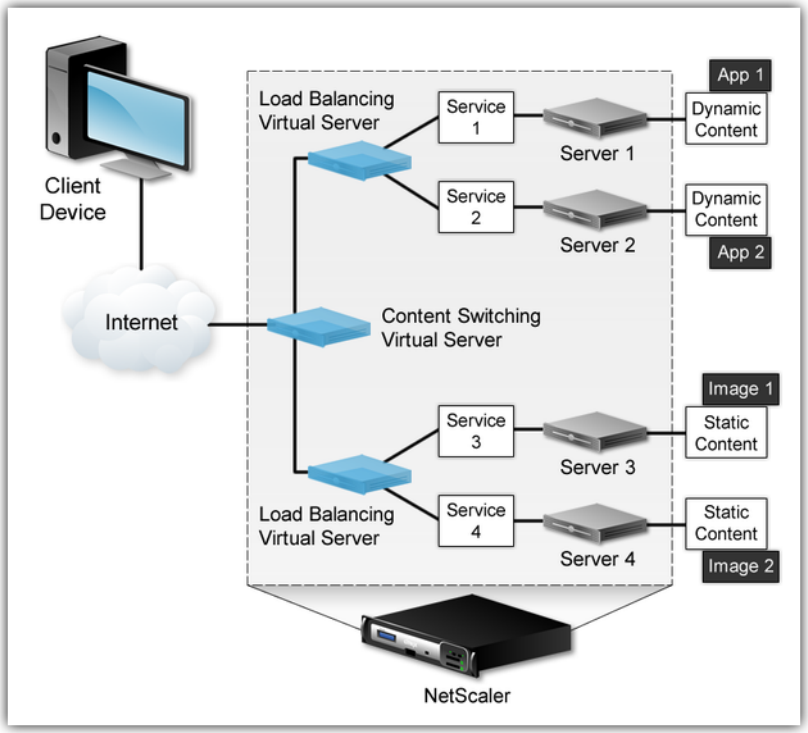
Understanding Content Switching

A typical content-switching configuration consists of a content-switching virtual server, content-switching policies and load-balancing virtual servers. Content switching is applicable to HTTP, HTTPS, and TCP traffic.

> SSL offloading needs to be enabled for HTTPS traffic.

When requests reach the content-switching virtual server, the NetScaler system applies the content-switching policies to them. The requests are then routed to the appropriate load-balancing virtual servers bound to the policies. The load-balancing virtual servers then send them to the services.

The content-switching feature allows the NetScaler system to replace application logic for directing traffic to servers. Content-switching virtual servers can send client requests only to other virtual servers. The figure shows the relationship between load-balancing virtual servers and content-switching virtual servers.



Content Switching Based on Layer Characteristics

The NetScaler system can also perform TCP content switching based on various characteristics, such as:

- Layer-3 characteristics, including:
  - Source IP address
  - Destination IP address
- Layer-4 characteristics, including:
  - TCP port
  - TCP maximum segment size (MSS) value
  - Buffered TCP payload

If the content-switching policy uses buffered TCP payload, the connection may stop responding if the client blocks the response of the server and the NetScaler system buffers the data.

The command syntax for TCP content switching is the same as HTTP content switching. You can configure layer-3-based and layer-4-based content switching using classic policies.

 © Copyright 2012 Citrix Systems, Inc.

Content-Switching Virtual Servers and Load-Balancing Virtual Servers

A content-switching virtual server is different from a load-balancing virtual server in the following ways:

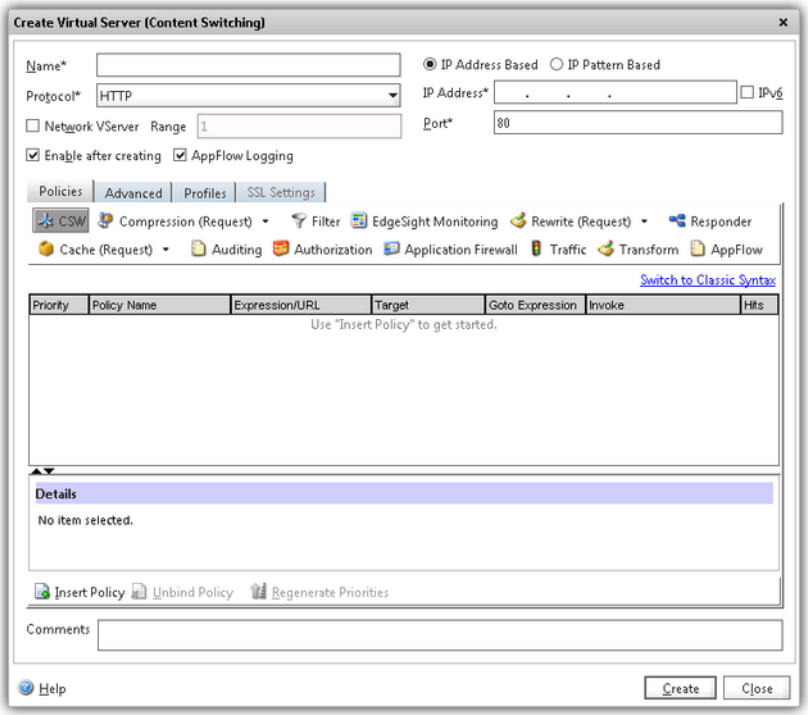
- You can configure the content-switching feature using classic or advanced policies, but not both.

> If classic policy expressions are bound to a virtual server, advanced policy expressions cannot be bound to the same virtual server.

- The content-switching virtual server does not directly address services; it points to load-balancing virtual servers. The load-balancing virtual servers associate the traffic with real services and servers.
- The process of distributing traffic among the associated load-balancing virtual servers is determined by the content-switching policy. On a load-balancing virtual server, this decision is made by the chosen load-balancing method.
- If the traffic does not match any content-switching policy, then the virtual server sends the traffic to a default load-balancing virtual server if one is defined.

> - If no default server is defined on the content-switching virtual server, the non-matched traffic is dropped.
> - The content-switching virtual server must perform SSL offload if the client is using SSL.

Configuring Content-Switching Virtual Servers



An essential part of configuring content switching is creating content-switching virtual servers. Content switching is available for:

- HTTP traffic
- HTTPS traffic
- Some TCP traffic

Virtual Server Configuration

Configuring a content-switching virtual server is very similar to configuring a load-balancing virtual server.

You configure a content-switching virtual server by defining a service, external IP address, and a port.

 © Copyright 2012 Citrix Systems, Inc.

Content-Switching Configuration Parameters

vServerName	Name of the content-switching virtual server. The name must not exceed 127 characters, and the leading character must be a number or letter.
IPAddress	IP address of the virtual server. This IP address (VIP) is usually a public IP address to which the clients send connection requests.
port	Port on which the virtual server listens for client connections. The port number must be between 0-65535.
protocol	Protocol of the requests processed by the content-switching virtual server.

To Add a Content-Switching Virtual Server Using the Configuration Utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, click **Add**.
3. In the Create Virtual Server (Content-Switching) dialog box, in the Name, IP Address, and Port text boxes, type the name, IP address, and port of the virtual server.

   If you need to enter an IPv6 address, select the IPv6 check box before you enter the address.
4. In the Protocol list, select the type of the virtual server.
5. Click **Create**, and then click **Close**.

Content-Switching Virtual Server Creation Using the Command-line Interface

At the NetScaler command prompt, type:

```
add cs vserver vServerName protocol IPAddress port
```

Example of NetScaler Command Prompt to Create a Content-Switching Virtual Server

```
add cs vserver Vserver-CS-1 HTTP 10.102.29.161 80
```

Test Your Knowledge

1. Content switching is applicable to which three types of traffic? (Choose three.)
    a. DNS
    b. FTP
    c. HTTP
    d. HTTPS
    e. TCP
2. Which two statements concerning how a content-switching virtual server differs from a load-balancing virtual server is true? (Choose two.)
    a. A content-switching virtual server is available only for HTTP/HTTPS traffic and is not implemented for any other protocol.
    b. If the traffic does not match any content-switching policy, the virtual server will send the traffic to a default load-balancing server if one is defined.
    c. A content-switching server references actual services.
    d. A content-switching virtual server can be configured using classic or advanced policies, but not both.

 © Copyright 2012 Citrix Systems, Inc.

Configure a Load-Balancing Setup for Content Switching

The content-switching virtual server redirects all requests to a load-balancing virtual server. You must create one load-balancing virtual server for each version of the content that is being switched. This is true even when your setup has only one server for each version of the content, and you are therefore not doing any load balancing with those servers. You can also configure actual load balancing with multiple load-balanced servers that mirror each version of the content. In either scenario, the content-switching virtual server needs to have a specific load-balancing virtual server assigned to each version of the content that is being switched.

The load-balancing virtual server then forwards the request to a service. If it has only one service bound to it, it selects that service. If it has multiple services bound to it, it uses its configured load-balancing method to select a service for the request, and forwards that request to the service that it selected.

To configure a basic load-balancing setup, you need to perform the following tasks:

- Create load-balancing virtual servers
- Create services
- Bind services to the load-balancing virtual server

Content-Switching Policies

A content-switching policy defines a type of request that is to be directed to a load-balancing virtual server. If you set a priority when binding your policies to the content-switching virtual server, the policies are evaluated in order of priority. If you do not set specific priorities when binding your policies, the policies are evaluated in the order in which they were created.

Domain-based policies	The NetScaler system compares the domain of an incoming URL with the domains specified in the policies. The system then returns the most appropriate content. Domain-based policies must be classic policies; default syntax policies are not supported for this type of content-switching policy.
URL-based policies	The system compares an incoming URL with the URLs specified in the policies. The system then returns the most appropriate URL-based content, which is usually the longest matching configured URL. URL-based policies must be classic policies; default syntax policies are not supported for this type of content-switching policy.
Rule-based policies	The system compares incoming data to expressions specified in the policies. You create rule-based policies by using either a classic expression or a default syntax expression. Both classic and default syntax policies are supported for rule-based content-switching policies.

 © Copyright 2012 Citrix Systems, Inc.

Binding Content-Switching Policies

After you create your content-switching virtual server and policies, you bind each policy to the content-switching virtual server. You can specify the target load-balancing virtual server for the policy or expression in the action parameter that the content-switching virtual server uses to determine the target load-balancing virtual server for forwarding the traffic.

To Bind a Policy to a Content-Switching Virtual Server and Select a Load-balancing Virtual Server Target Using the Configuration Utility

1. In the navigation pane, expand **Content Switching**, and then click **Virtual Servers**.
2. In the details pane, double-click the virtual server for which you want to bind the policy.
3. In the Configure Virtual Server (Content-Switching) dialog box, on the **Policies** tab, click **Insert Policy**. A list of existing policies appears in the Policy Name drop-down list. You can either select an existing policy or you can create a new policy.
4. Click **OK**.

Policies Bound to a Content-Switching Virtual Server and Selection of a Load-Balancing Virtual Server Target Using the Command-line Interface

At the NetScaler command prompt, type:

```
bind cs vserver ContentSwitchingVirtualServerName
TargetLoadBalancingVirtualServerName -policyname PolicyName -
priority
PositiveInteger -gotoPriorityExpression expression -
type ( REQUEST | RESPONSE )
-invoke (labelType labelName)
```

Examples of NetScaler Command Prompt to Bind a Policy to a Content-Switching Virtual Server and Selection of a Load-Balancing Virtual Server Target

```
bind cs vserver csw-vip2 -policyname csw-ape-policy2 -
priority 14 -gotoPriorityExpression NEXT
```

```
bind cs vserver csw-vip3 -policyname rewrite-policy1 -
priority 17 -gotoPriorityExpression 'q.header("a").count' -
flowtype REQUEST -invoke policylabel label1
```

© Copyright 2012 Citrix Systems, Inc.

```
bind cs vserver Vserver-CS-1 Vserver-LB-1 -policyname Policy-CS-
1 -priority 20
```

 © Copyright 2012 Citrix Systems, Inc.

Content-Switching Rule Precedence Without Priority Specified

A rule precedence often follows the order in which the rules are configured. If no priority is used when binding the policy, the policy is assigned a priority of 0, making it the highest priority. If there is more than one policy with the same priority, they are evaluated in the order that they are created.

```
bind cs vserver cswvip lbvip1 -policy cs_policy_a
bind cs vserver cswvip lbvip2 -policy cs_policy_b
bind cs vserver cswvip lbvip3 -policy cs_policy_c
```

Evaluation order:

1. cs_policy_a
2. cs_policy_b
3. cs_policy_c

© Copyright 2012 Citrix Systems, Inc.

Content-Switching Rule Precedence With Priority Specified

The `priority` parameter can be configured when the rule is bound to the virtual server using a policy. If a priority is provided, the rules are processed from lowest priority value to highest, as shown in the following example:

```
bind cs vserver cswvip lbvip1 -policy cs_policy_a -priority 10
```

```
bind cs vserver cswvip lbvip2 -policy cs_policy_b -priority 20
```

```
bind cs vserver cswvip lbvip3 -policy cs_policy_c -priority 15
```

Evaluation order:

1. cs_policy_a
2. cs_policy_c
3. cs_policy_b

Rule-Based Policy Example

The following command-line interface commands create two rule-based content-switching policies:

```
add cs policy match_big_bad_or_long -rule big_bad_or_long
```

```
add cs policy html_w_cookie -
rule "accepts_html && has_cookie_header"
```

The first command defines a policy as matching a rule that is a compound expression. The –rule parameter links an expression to the policy. The second command creates a policy based on a compound expression that is defined as part of the policy statement. The effect of the two policies is to define two different types of traffic: one with characteristics of malformed URLs and the other with a classification of browsers with a specific cookie present.

The following command-line interface commands bind the policies to the content-switching virtual server:

```
bind cs vserver cswvip honeypot_lbvip -
policy match_big_bad_or_long
```

```
bind cs vserver cswvip normal_lbvip -policy html_w_cookie
```

```
bind cs vserver cswvip default_lbvip
```

 © Copyright 2012 Citrix Systems, Inc.

The content-switching server is bound to these policies so that the potentially malicious traffic is directed to a "honeypot" server, which is a server designed to accept and analyze hostile traffic without exposing valuable resources. The cookie-enabled browsers are directed to the existing customer site, and all other traffic is directed to a default server.

Unmatched Traffic Handling

If a default group is configured for the content-switching virtual server, then the request is forwarded to the configured default group. If the configured default group is down, or no default group is configured, an HTTP 404 Not Found error message is sent to the client.

Test Your Knowledge

1. You must create one load balancing virtual server for each version of the content that is being switched unless your setup has only one server for each version of the content and you are not doing any load balancing with those servers.
    a. True
    b. False
2. Which content-switching policy supports both classic and default syntax policies?
    a. Domain-based policies
    b. Rule-based policies
    c. URL-based policies

 © Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

Put the following policies in the order that they will be evaluated.

Order of Evaluation	Description
	bind cs vserver cswvip lbvip3 -policy cs_policy_c -priority 15
	bind cs vserver cswvip lbvip3 -policy cs_policy_c -priority 20
	bind cs vserver cswvip lbvip3 -policy cs_policy_c -priority 10

© Copyright 2012 Citrix Systems, Inc.

© Copyright 2012 Citrix Systems, Inc.

Module 7

Connection Tuning

© Copyright 2012 Citrix Systems, Inc.

Overview

Once you have configured the NetScaler for networking and load balancing, there are several enhancements that you should consider to fine tune the performance of the NetScaler in your environment.

If your environment uses IPv6, the NetScaler system supports both server-side and client-side IPv6 and can therefore function as an IPv6 node. IPv6 support is a licensed feature, which you have to enable before you can use or configure it. You can also configure the NetScaler to function as an ADNS server, DNS proxy server, end resolver, and forwarder. The NetScaler also supports both dynamic and static routing.

For environments that require more than basic load balancing, there are many ways to enhance your load-balancing configuration, including link load balancing, advanced load balancing methods, and using persistence groups. You should also plan to protect and manage your load-balancing configuration.

After completing this module, you will be able to:

- Explain how the NetScaler supports IPv6.
- Explain how to configure the NetScaler to function as an ADNS server, DNS proxy server, end resolver, and forwarder.
- Explain when to use the NetScaler to support dynamic and static routing.
- Describe when Link Load Balancing should be configured.
- Identify when to configure persistence on a load-balancing virtual server.
- Describe how to protect your load-balancing virtual server against failure.
- Identify how to manage your load balancing setup.

© Copyright 2012 Citrix Systems, Inc.

IPv6

The NetScaler system supports both server-side and client-side IPv6 and can therefore function as an IPv6 node. It can accept connections from IPv6 nodes, both hosts and routers, and from IPv4 nodes, and can perform Protocol Translation (RFC 2765) before sending traffic to the services.

IPv6 support is a licensed feature, which you have to enable before you can use or configure it. If IPv6 is disabled, the NetScaler does not process IPv6 packets. It displays a warning when you run an unsupported command. It will also display a warning if you attempt to run IPv6 commands without the appropriate license.

For more information about IPv6, see Citrix eDocs at *http://edocs.citrix.com*.

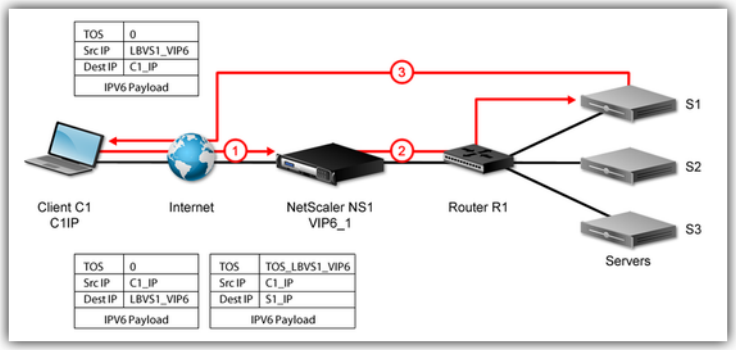
Load Balancing in Direct Server Return Mode for IPv6 Networks

Differentiated services (DS), also known as Type of Service (TOS), is a field that is part of the TCP packet header. TOS is used by upper layer protocols for optimizing the path for a packet. The TOS information encodes the NetScaler system virtual IP address (VIP), and the load balanced servers extract the VIP from it.

You can configure load balancing in Direct Server Return (DSR) mode for IPv6 networks by using the Type of Service (TOS) field when the NetScaler system and the servers are in different networks. In DSR mode, when a client sends a request to a VIP6 address on a NetScaler system, the system forwards this request to the server by changing the destination IPv6 address of the packet to the IPv6 address of the server and sets an encoded value of the VIP6 address in the TOS (also called traffic class) field of the IPv6 header. You can configure the server to use the information in the TOS field to derive the VIP6 address from the encoded value, which is then used as source IP address in response packets. Response traffic directly goes to the client, bypassing the NetScaler.

The NetScaler system implements IP tunneling by encapsulating data packets that it receives. The encapsulation adds header information. The system then forwards the encapsulated data packets to the router, or appropriate server, using tunnels. The NetScaler can also act as a decapsulator if placed in front of the load-balanced server.

 © Copyright 2012 Citrix Systems, Inc.



In the image above, traffic flows in the following order:

1. Client C1 sends a request to the virtual server LBVS1.
2. The load-balancing algorithm for LBVS1 selects server S1 and the NetScaler system opens a connection to S1. NS1 sends the request to S1 with:
   - TOS field set to TOS_LBVS1_VIP6.
   - Source IP address as C1_IP.
3. After receiving the request, the server S1 uses the information in the TOS field to derive the LVBS1_VIP6 address, which is the IP address of the virtual server LBVS1 on NS1. The server directly sends the response to C1, bypassing the NetScaler. The response contains the Source IP address set to the derived LBVS1_VIP6 address so the client communicates to the virtual server LVBS1 on NS1 and not to server S1.

VLAN as Gateway in the IPv6 Routes

You can specify a VLAN instead of a gateway while adding IPv6 static routes. This option is required for adding directly connected IPv6 routes.

Also, the NetScaler system supports adding directly connected routes, discovered by the dynamic routing protocols, that include a VLAN ID instead of a gateway.

Reverse Network Address Translation for IPv6 Traffic

You can add Reverse Network Address Translation (RNAT) rules for IPv6 packets. These RNAT rules are called RNAT6s. When an IPv6 packet generated by the server matches the conditions specified in the RNAT6 rule, the system replaces the source IPv6 address of the IPv6 packet with a configured NAT IPv6 address before forwarding it to the destination. The NAT IPv6 address must be one of the NetScaler-owned SNIP6 or VIP6 addresses.

When configuring an RNA6 rule, you can specify either an IPv6 prefix or an ACL6 as the condition:

- When you use an IPv6 prefix, the system performs RNAT processing on those IPv6 packets whose IPv6 address matches the prefix.
- When you use an ACL6, the system performs RNAT processing on those IPv6 packets that match the conditions specified in the ACL6.

 © Copyright 2012 Citrix Systems, Inc.

DNS and the NetScaler

You can configure the NetScaler system to function as an ADNS server, DNS proxy server, end resolver, and forwarder. You can add DNS resource records on the NetScaler, including service (SRV) records; IPv6 (AAAA) records; address (A) records; mail exchange (MX) records; canonical name (CNAME) records; pointer (PTR) records; start of authority (SOA) records; and text (TXT) records. Also, you can configure the NetScaler to load balance external DNS name servers.

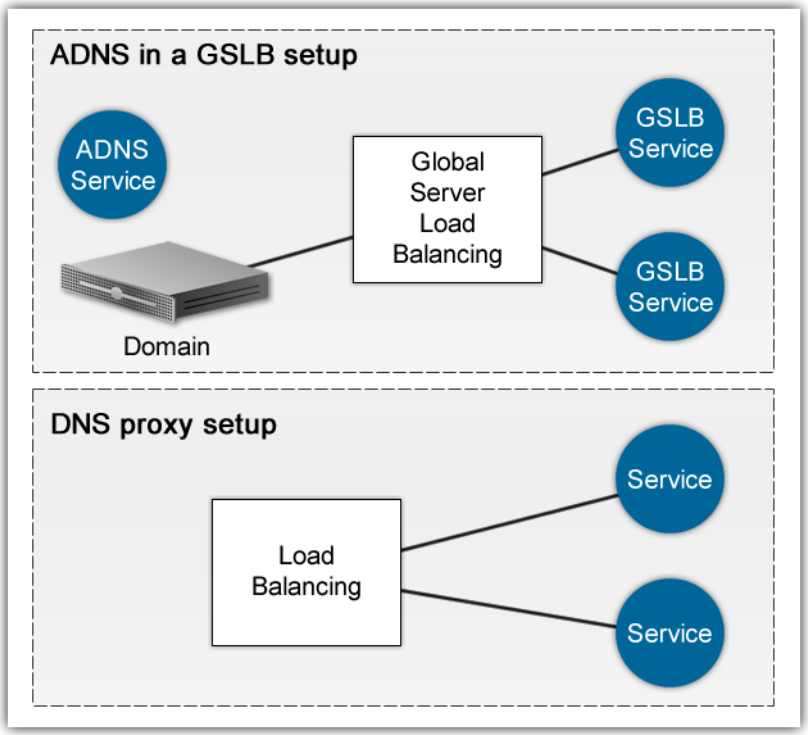
The NetScaler can be configured as the authority for a domain. To do this, you add valid SOA and NS records for the domain.

To configure the NetScaler as an ADNS server for a zone, you must add an ADNS service, and then configure the zone. To do so, you add valid SOA and NS records for the domain. When a client sends a DNS request, the NetScaler system searches the configured resource records for the domain name. You can configure the ADNS service to be used with the NetScaler global server load balancing (GSLB) feature.

You can delegate a subdomain by adding NS records for the subdomain to the zone of the parent domain. You can then make the NetScaler authoritative for the subdomain by adding a "glue record" for each of the subdomain name servers. If GSLB is configured, the NetScaler makes a GSLB decision based on its configuration and replies with the IP address of the selected virtual server. The following figure shows the entities in an ADNS GSLB setup and a DNS proxy setup.

The NetScaler system can function as a DNS proxy. Caching of DNS records, which is an important function of a DNS proxy, is enabled by default on the NetScaler system. This enables the NetScaler to provide quick responses for repeated translations. You must also create a load balancing DNS virtual server and DNS services and then bind these services to the virtual server.

© Copyright 2012 Citrix Systems, Inc.



NetScaler as an ADNS Server

You can configure the NetScaler system to function as an authoritative domain name server (ADNS) for a domain. As an ADNS server for a domain, the NetScaler resolves DNS requests for all types of DNS records that belong to the domain. To configure the NetScaler to function as an ADNS server for a domain, you must create an ADNS service and configure NS and Address records for the domain on the NetScaler. Normally, the ADNS service uses the Mapped IP address (MIP). However, you can configure the ADNS service with any NetScaler-owned IP address.

During DNS resolution, the ADNS server directs the DNS proxy or local DNS server to query the NetScaler for the IP address of the domain. Because the NetScaler is authoritative for the domain, it sends the IP address to the DNS proxy or local DNS server.

In ADNS mode, removal of SOA and ADNS records also removes functionality for the following items in the domain hosted by the NetScaler:

- ANY query
- NODATA, NXDOMAIN, and other negative responses

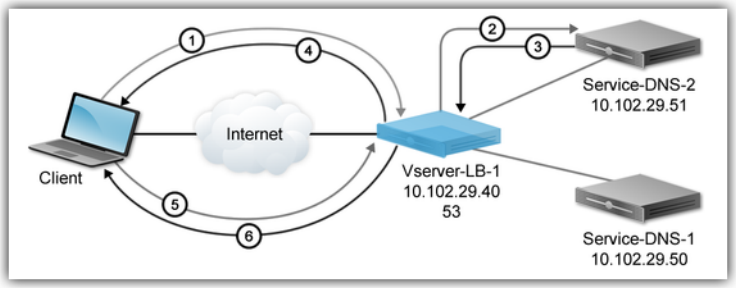
 © Copyright 2012 Citrix Systems, Inc.

NetScaler as a DNS Proxy Server

As a DNS proxy server, the NetScaler system can function as a proxy for either a single DNS server or a group of DNS servers. By default, the NetScaler system caches responses from DNS name servers. When the system receives a DNS query, it checks for the queried domain in its cache. If the address for the queried domain is present in its cache, the NetScaler returns the corresponding address to the client. Otherwise, it forwards the query to a DNS name server that checks for the availability of the address and returns it to the NetScaler. The NetScaler then returns the address to the client.

For requests for a domain that has been cached earlier, the NetScaler serves the Address record of the domain from the cache without querying the configured DNS server.

The NetScaler discards a record stored in its cache when the time-to-live (TTL) value of the record reaches the configured value. A client that requests an expired record has to wait until the NetScaler retrieves the record from the server and updates its cache. To avoid this delay, the NetScaler proactively updates the cache by retrieving the record from the server before the record expires.



NetScaler as a DNS End Resolver

A resolver is a procedure that is invoked by an application program that translates a domain/host name to its resource record. The resolver interacts with the LDNS, which looks up the domain name to obtain its IP address. The NetScaler can provide end-to-end resolution for DNS queries.

In recursive resolution, the NetScaler system queries different name servers recursively to access the IP address of a domain. When the NetScaler receives a DNS request, it checks its cache for the DNS record. If the record is not present in the cache, it queries the root servers configured in the ns.conf file. The root name server reports with the address of a DNS server that has detailed information about the second-level domain. The process is repeated until the required record is found.

When you start the NetScaler system for the first time, 13 root name servers are added to the ns.conf file. The NS and Address records for the 13 root servers are also added. You can modify the ns.conf file, but the NetScaler does not allow you to delete all 13 records; at least one name server entry is required for the system to perform name resolution. The following diagram illustrates the process of name resolution.

© Copyright 2012 Citrix Systems, Inc.

For recursive resolution functionality, caching should be enabled.

NetScaler as a DNS Forwarder

A forwarder is a server that forwards DNS queries to DNS servers that are outside the forwarder server's network. Queries that cannot be resolved locally are forwarded to other DNS servers. A forwarder accumulates external DNS information in its cache as it resolves DNS queries. To configure the NetScaler as a forwarder, you must add an external name server (a name server other than the NetScaler system).

The NetScaler system allows you to add external name servers to which it can forward the name resolution queries that cannot be resolved locally. To configure the NetScaler system as a forwarder, you must add the name servers where the system should forward name resolution queries. You can specify the lookup priority to specify the name service that the NetScaler system must use for name resolution.

 © Copyright 2012 Citrix Systems, Inc.

IP Routing

NetScaler systems support both dynamic and static routing. Because simple routing is not the primary role of a NetScaler, the main objective of running dynamic routing protocols is to enable route health injection (RHI), so that an upstream router can choose the best among multiple routes to a topographically distributed virtual server.

Most NetScaler implementations use some static routes to reduce routing overhead. You can create backup static routes and monitor routes to enable automatic switchover in the event that a static route goes down. You can also assign weights to facilitate load balancing among static routes, create null routes to prevent routing loops, and configure IPv6 static routes. You can configure policy-based routes (PBRs), for which routing decisions are based on criteria that you specify.

Static Routes

Static routes are manually created to improve the performance of your network. You can monitor static routes to avoid service disruptions. Also, you can assign weights to ECMP routes, and you can create null routes to prevent routing loops.

The NetScaler supports monitoring of IPv4 and IPv6 static routes. You can configure the NetScaler to monitor an IPv4 static route either by creating a new ARP or PING monitor or by using existing ARP or PING monitors. You can configure the NetScaler to monitor an IPv6 static route either by creating a new Neighbor discovery for IPv6 (ND6) or PING monitor or by using the existing ND6 or PING monitors.

Policy-based Routing

Use cases for policy-based routing are:

- Traffic originating from different network with different routing requirement, such as selecting a different next hop
- Secure or non-secure links for different types of originating traffic
- Traffic isolation achieved for environments with shared infrastructure, such as those using different routes for different virtual IP addresses, would simulate the multitenant deployments of service providers

An example of policy-based routing is when a policy redirects traffic from an internal IP range to an internal intranet site, while routing Internet traffic to a public site. These sites would exist on different networks and would not be virtual servers and services configured on the NetScaler system, as the NetScaler system supplies the next hop and not the destination of the request.

In multitenancy environments, the same IP addresses might be configured for different users of the same system. To cope with these complexities, use policy-based routing to direct traffic differently based upon the source or destination IP addresses.

© Copyright 2012 Citrix Systems, Inc.

Policy-based Routing Parameters

Policy-based routing provides a way to select the next hop based on the following parameters:

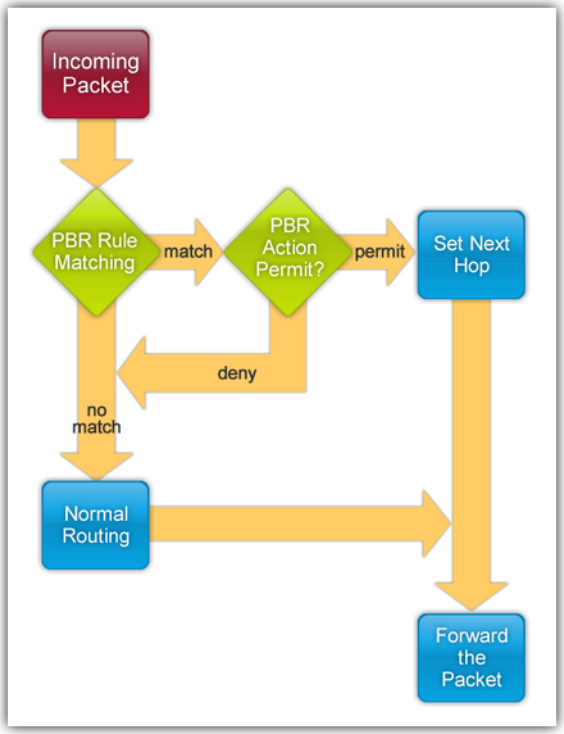
- Source IP address
- Source port
- Destination IP address
- Destination port
- Protocol
- VLAN
- Interface
- MAC address

Policy-Based Action Configuration

The configuration process for policy-based routing is similar to access-control-list configuration. If there is an ALLOW action, then the next hop is invoked. If there is a DENY, the packet is routed normally.

This figure shows the process flow for how policy-based routing is evaluated and the resulting action. Resulting actions include changing the next hop as configured, or routing as normal.

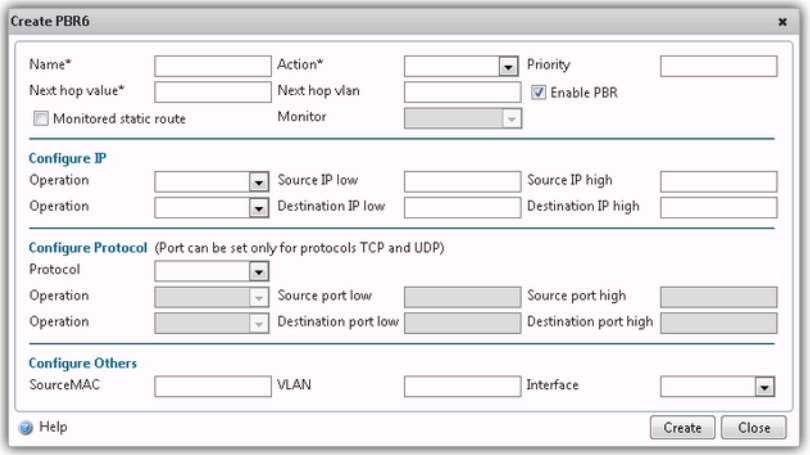
 © Copyright 2012 Citrix Systems, Inc.



Policy-based Routes for IPv6 Traffic

You can add policy-based routes for outgoing IPv6 packets. Policy-based routing bases routing decisions on criteria that you specify. An IPv6 policy-based route (PBR6) specifies criteria for selecting IPv6 packets and, typically, a next hop destination to send the selected packets. For example, you can configure the NetScaler system to route outgoing IPv6 packets from a specific IPv6 address or range to a particular next hop router. Each packet is matched against each configured PBR6, in the order determined by the specified priorities, until a match is found. If no match is found, or if the matching PBR6 specifies a DENY action, the NetScaler applies the routing table for normal destination-based routing.

If the Next hop value field is a link local address, then you should specify a destination VLAN ID to be used.



 © Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

Match the following NetScaler functions with the correct descriptions.

- ADNS Server
- DNS End Resolver
- DNS Forwarder
- DNS Proxy Server

Description	Term
The NetScaler resolves DNS requests for all types of DNS records that belong to the domain by directing the DNS proxy or local DNS server to query the NetScaler for the IP address of the domain.	
When the system receives a DNS query, it checks for the queried domain in its cache and, for requests that have been cached earlier, the NetScaler serves the address record of the domain from the cache without querying the configured DNS server.	
This is invoked by an application program that translates a domain/host name to its resource record. It interacts with the LDNS, which looks up the domain name to obtain its IP address.	
This server forwards DNS queries that cannot be resolved locally to other DNS servers outside the network.	

© Copyright 2012 Citrix Systems, Inc.

Configuration of Load Balancing for Commonly-Used Protocols

In addition to web sites and web-based applications, other types of network-deployed applications that use other common protocols often receive large amounts of traffic and therefore benefit from load balancing. Several of these protocols require specific configurations for load balancing to work properly. Among them are FTP, DNS, SIP, and RTSP.

For more information about specific configurations for load balancing, see Citrix article CTX132359 at *http://support.citrix.com*.

 © Copyright 2012 Citrix Systems, Inc.

Link Load Balancing

Link load balancing (LLB) balances outbound traffic across multiple Internet connections provided by different service providers. LLB enables the NetScaler system to monitor and control traffic so that packets are transmitted seamlessly over the best possible link. Unlike with server load balancing, where a service represents a server, with LLB, a service represents a router or the next hop. A link is a connection between the NetScaler and the router.

Load-balancing methods that are applicable to LLB are:

- Round robin
- Destination IP hash
- Least bandwidth
- Least packets

You can optionally configure persistence for connections to be sustained on a specific link. The available persistence types are:

- Source IP address-based
- Destination IP address-based
- Source IP and destination IP address-based

PING is the default monitor but configuring a transparent monitor is recommended.

For more information about about configuring link load balancing, see Citrix article CTX32359 on *http://support.citrix.com*.

IPv6 Support for Link Load Balancing

The NetScaler LLB feature supports IPv6 addresses. This support is required when Internet service providers assign IPv6 addresses to customers. Configuring an LLB setup with IPv6 addresses is similar to configuring a setup with IPv4 addresses, except that you now create an IPv6 service to represent your router or the next hop. Three new CLI commands (`add lb route6` , `show lb route6` , and `rm lb route6`) have been added for configuring an IPv6 route. The configuration utility has a new LLBv6 tab in the **Network** > **Routes** pane.

SIP Load Balancing

The NetScaler system can load balance SIP servers to improve the performance of a VoIP system.

To configure a basic load-balancing setup for SIP, you create services and virtual servers of type SIP-UDP, naming the entities and setting the parameters. You must then configure RNAT.

For more information about SIP load balancing and configuring RNAT, see Citrix article CTX132359 on *http://support.citrix.com*.

 © Copyright 2012 Citrix Systems, Inc.

Advanced Load-Balancing Methods

Three advanced load-balancing methods that are available on the NetScaler system are least-response-time based on monitoring, token-based load balancing, and hash load balancing. The least-response-time method selects the service with the fewest active connections and the lowest average response time. Token load balancing is based on matching a token with the request or TCP payload. Hash load balancing includes several methods that use a hash value of some aspect of the client request to determine where to forward the request.

Least-Response-Time

When the load-balancing virtual server is configured to use the least-response-time method, it selects the service with the fewest active connections and the lowest average response time. You can configure this method for HTTP and Secure Sockets Layer (SSL) services only. The response time (also called Time to First Byte, or TTFB) is the time interval between sending a request packet to a service and receiving the first response packet from the service. The NetScaler system uses response code 200 to calculate TTFB.

Before you use the least-response-time method with monitoring, you must bind application-specific monitors to each service and enable least-response-time method mode on these monitors. The NetScaler system then makes load balancing decisions based on the response times it calculates from monitoring probes.

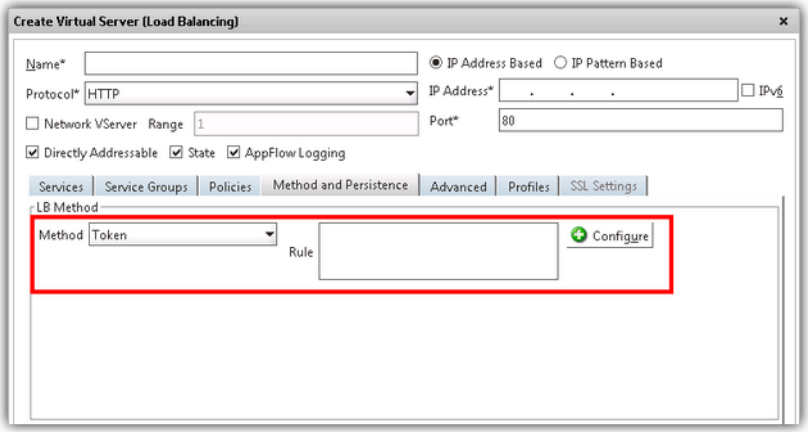
You can use the least-response-time method with monitors to select non-HTTP and non-HTTPS services. You can also use this method when several monitors are bound to a service. Each monitor determines the response time by using the protocol that it measures for the service that it is bound to. The virtual server then calculates an average response time for that service by averaging the results.

Token

A token is created based on the client request, and the request is forwarded to a service based on the results of that token. The exact choice of variables and token configuration are end-user configurable. An advantage of this kind of load-balancing method is that it works across different virtual servers. For example, it ensures that requests for HTTP and FTP are served from the same physical server, even though they are advertised through two different virtual servers.

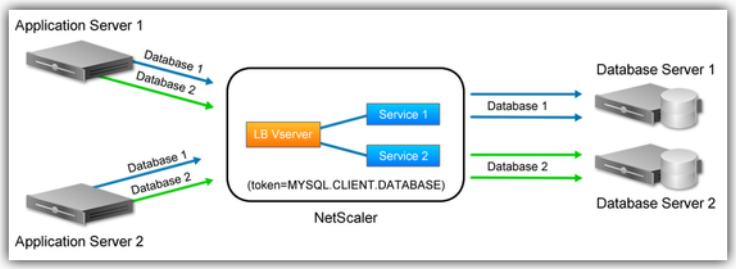
© Copyright 2012 Citrix Systems, Inc.

Token Method for Load-Balancing DataStream Example



You can configure the token method of load balancing for DataStream to base the selection of database servers on the value of the token extracted from the client (application or web servers) requests. These tokens are defined by using SQL expressions. For subsequent requests with the same token, the NetScaler sends the requests to the same database server that handled the initial request. Requests with the same token are sent to the same database server until the maximum connection limit is reached or until the session entry has expired. Once the maximum connection limit is reached, a new server is selected based on the token and remaining servers.

Token Method for Load-Balancing Architecture



Hashing

Hashing comprises a number of different methods that all use a hash value of a particular aspect of the client request to determine where to forward the request. These hashing algorithms ensure minimal disruption when services are added to or deleted from your load-balancing setup. Most of them calculate two hash values:

 © Copyright 2012 Citrix Systems, Inc.

- A hash of the service's IP address and port.
- A hash of the incoming URL, the domain name, the source IP address, the destination IP address, or the source and destination IP addresses, depending on the configured hash method.

The NetScaler system then generates a new hash value by using both of those hash values. Finally, it forwards the request to the service with highest hash value. As the system computes a hash value for each request and selects the service that will process the request, it populates a cache. Subsequent requests with the same hash value are sent to the same service.

Hashing methods allow the load balancer to direct similar requests to the same physical resource. Depending on the type of hashing, the similarity of the requests can be controlled. For example, source IP address hashing allows clients from the same subnets to be directed to the same service.

The NetScaler hashing types include:

- URL hash method
- Domain hash method
- Destination IP hash method
- Source IP hash method
- Source IP Destination IP hash method
- Source IP Source Port hash method
- Call ID hash method

For more information about hashing methods, see Citrix article CTX132359 at *http://support.citrix.com*.

The URL Hash Method

When you configure the NetScaler system to use the URL hash method for load balancing the services, the NetScaler generates a hash value of the HTTP URL present in the incoming request. If the service selected by the hash value is DOWN, the algorithm has a method to select another service from the list of active services. The NetScaler caches the hashed value of the URL, and when it receives subsequent requests that use the same URL, it forwards them to the same service. If the NetScaler cannot parse an incoming request, it uses the round robin method for load balancing instead of the URL hash method.

For generating the hash value, NetScaler uses a specific algorithm and considers a part of the URL. By default, the NetScaler considers the first 80 bytes of the URL. If the URL is of less than 80 bytes, the complete URL is used. You can specify a different length. The hash length can be from 1 to 4096 bytes. Generally, if long URLs are used where only a small number of characters are different, it is a good idea to make the hash length as high as possible to try to ensure a more even load distribution.

© Copyright 2012 Citrix Systems, Inc.

The Domain Hash Method

A load-balancing virtual server configured to use the domain hash method uses the hashed value of the domain name in the HTTP request to select a service. The domain name is taken from either the incoming URL or the Host header of the HTTP request. If the domain name appears in both the URL and the Host header, the NetScaler gives preference to the URL.

If you configure domain name hashing, and an incoming HTTP request does not contain a domain name, the NetScaler system defaults to the round robin method for that request.

The hash-value calculation uses the name length or hash length value, whichever is smaller. By default, the NetScaler system calculates the hash value from the first 80 bytes of the domain name. To specify a different number of bytes in the domain name when calculating the hash value, you can set the `hashLength` parameter (Hash Length in the configuration utility) to a value of from 1 to 4096 bytes.

The Destination IP Hash Method

A load-balancing virtual server configured to use the destination IP hash method uses the hashed value of the destination IP address to select a server. You can mask the destination IP address to specify which part of it to use in the hash-value calculation, so that requests that are from different networks but destined for the same subnet are all directed to the same server. This load-balancing method is appropriate for use with the cache redirection feature.

The Source IP Hash Method

A load-balancing virtual server configured to use the source IP hash method uses the hashed value of the client IP address to select a service. To direct all requests from source IP addresses that belong to a particular network to a specific destination server, you must mask the source IP address.

The Source IP Destination IP Hash Method

A load-balancing virtual server configured to use the source IP destination IP hash method uses the hashed value of the source and destination IP addresses to select a service. Hashing is symmetric; the hash-value is the same regardless of the order of the source and destination IPs. This ensures that all packets flowing from a particular client to the same destination are directed to the same server. To direct all requests that belong to a particular network to a specific destination server, you must mask the source IP address.

The Source IP Source Port Hash Method

A load-balancing virtual server configured to use the source IP source port hash method uses the hash value of the source IP and source port to select a service. This ensures that all packets on a

 © Copyright 2012 Citrix Systems, Inc.

particular connection are directed to the same service. This method is used in connection mirroring and firewall load balancing.

The Call ID Hash Method

A load-balancing virtual server configured to use the call ID hash method uses the hash value of the call ID in the SIP header to select a service. Packets for a particular SIP session are therefore always directed to the same proxy server. This method is applicable to SIP load balancing.

© Copyright 2012 Citrix Systems, Inc.

Custom Load

The custom load balancing method calculates load by using metrics that have been created and bound to a metric table. Once a metric has been created and bound, the metric table is then associated with a load monitor. Metrics can then be bound to the monitor from the pool of available metrics that reside in the metric table.

The custom load-balancing method distributes traffic based on the load calculations made by load monitors. The service with the least load value, as calculated by the monitors, is chosen to handle the client requests.

All services bound to a virtual server for which the load-balancing method is custom load must have load monitors bound to them. If no load monitors are bound, the virtual server will use the round-robin load-balancing method.

The local metric table is implicitly bound to the monitor and no additional binding operation is required.

For more information about custom load balancing, see Citrix article CTX132359 at *http://support.citrix.com*.

 © Copyright 2012 Citrix Systems, Inc.

HTTP-ECV and TCP-ECV Monitors

Monitors of type HTTP-ECV and TCP-ECV may be configured to check that the response to a specific request from the NetScaler to the Web server contains a response containing particular data. When the response containing the particular data is received, the NetScaler presumes that the site is up. This may be required in scenarios where simpler monitoring does not provide an adequate guarantee that the server is actually functioning sufficiently to serve the requests. For example, a monitor may be defined to probe a Web server that is a front end to a database, and the response should be checked to ensure that the Web server is querying the database correctly.

HTTP-ECV Monitors

The NetScaler establishes a TCP connection with the monitor destination. When the connection is established, the NetScaler sends the HTTP data specified by the -send parameter to the service and expects the HTTP response that the -receive parameter specifies. (HTTP body part, not including HTTP headers.) Empty response data matches any response. Expected data may be anywhere in the first 24K bytes of the HTTP body of the response.

- send [""] - HTTP data that is sent to the service
- recv [""] - the expected HTTP response data from the service

TCP-ECV Monitors

The NetScaler appliance establishes a 3-way handshake with the monitor destination. When the connection is established, the appliance uses the send parameter to send specific data to the service and expects a specific response through the receive parameter.

- send [""] - is the data that is sent to the service. The maximum permissible length of the string is 512 K bytes.
- recv [""] – expected response from the service. The maximum permissible length of the string is 128 K bytes.

© Copyright 2012 Citrix Systems, Inc.

Persistence and Persistence Connections

Unless you configure persistence, a load-balancing stateless protocol, such as HTTP, disrupts the maintenance of state information about client connections. Different transmissions from the same client might be directed to different servers even though all of the transmissions are part of the same session. You must configure persistence on a load-balancing virtual server that handles certain types of web applications, such as shopping cart applications.

Before you can configure persistence, you need to know the difference between the different types of persistence, how they are used, and what the implications of each type is. You then need to configure the NetScaler system to provide persistent connections for those web sites and web applications that require them.

You can also configure backup persistence, which takes effect in the event that the primary type of persistence configured for a load-balancing virtual server fails. You can configure persistence groups so that a client transmission to any virtual server in a group can be directed to a server that has received previous transmissions from the same client.

Session Persistence Method Overview

Session persistence methods are determined based on the method you assign to the service. The following table provides an overview of the most commonly used persistence methods.

Persistence Method	Description
Cookie-Insert Persistence	Cookie-insert persistence inserts an HTTP cookie in the client responses. The cookie is then used to direct subsequent requests from the client to the same physical server that handled the initial request.
Source-IP-Address Persistence	Source-IP-address persistence routes a client request based on the configured load-balancing method, and then directs all subsequent requests from the same IP address to the physical server that handled the initial request.
SSL Session ID Persistence	SSL-session-ID persistence is based on the arriving SSL session ID in a request, which is part of the SSL handshake process. Requests containing the same SSL session ID are directed to the physical server that processed the initial request.

 © Copyright 2012 Citrix Systems, Inc.

Persistence Method	Description
URL Passive Persistence	URL-passive persistence looks for an embedded server ID in client requests and then directs the request to the physical server that corresponds to the ID.
Custom Server ID Persistence	Custom-server-ID persistence looks for an embedded server ID in client requests and then directs the requests to the physical server that corresponds to the ID.
Rule-based Persistence	Rule-based persistence evaluates a client request against a configured rule and creates a persistence session if the request is a match.
Destination IP Address Persistence	Destination-IP-address persistence is used with link load balancing. When you use this persistence method, the system selects a service based on the configured load-balancing method and then directs all subsequent packets that are destined to the same IP address as the first packet to the same service.
Source/Destination IP Address Persistence	Source/destination-IP-address persistence is used with intrusion-detection-system (IDS) load balancing. When you use this persistence method, the system selects a service based on the configured load-balancing method and then directs all subsequent packets traveling from the same source to the same destination to the same physical service that received the first packet.
SIP Call ID Persistence	With SIP Call ID persistence, the NetScaler system chooses a service based on the call ID in the SIP header. This enables it to direct packets for a particular SIP session to the same service and, therefore, to the same load balanced server. This persistence type is applicable specifically to SIP load balancing.

© Copyright 2012 Citrix Systems, Inc.

Persistence Method	Description
RTSP Session ID	With RTSP Session ID persistence, when the NetScaler system receives a request from a new client, it creates a new persistence session based on the Real-Time Streaming Protocol (RTSP) session ID in the RTSP packet header, and then directs the request to the RTSP service selected by the configured load-balancing method.

Persistence Tables

Session persistence information for each session is stored on the NetScaler system in a persistence table. The following information is displayed in the table.

- Persistence type
- Source IP address
- Destination IP address
- Destination port
- Virtual server name
- Persistent session timeout
- Reference count
- SIP CALLID

Persistence Group Configuration

When you have load-balanced servers that handle several different types of connections (such as web servers that host multimedia), you can configure a virtual server group to handle these connections. To create a virtual server group, you bind different types of virtual servers, one for each type of connection that your load balanced servers accept, into a single group. You then configure a persistence type for the entire group.

You can configure either source-IP-based persistence or HTTP cookie-based persistence for persistence groups. After you set persistence for the entire group, you cannot change it for individual virtual servers in the group. If you configure persistence on a group and then add a new virtual server to the group, the persistence of the new virtual server is changed to match the persistence setting of the group.

When persistence is configured on a group of virtual servers, persistence sessions are created for initial requests, and subsequent requests are directed to the same service as initial request, regardless of the virtual server in the group that receives each client request.

For more information about creating and modifying a virtual persistency group, see Citrix article CTX132359 at *http://support.citrix.com*.

 © Copyright 2012 Citrix Systems, Inc.

Viewing of Persistence Sessions

You can view the different persistence sessions that are in effect globally or for a particular virtual server.

To View Persistence Session Using the Configuration Utility

1. In the navigation pane, click **Load Balancing**.
2. In the details pane, under **Monitor Sessions**, click **Virtual Server persistence sessions**.

Persistence Session Viewing Using the Command-line Interface

At the NetScaler command line, to view all persistence sessions type:

```
show lb persistentSessions <vServer>
```

Example of NetScaler Command Prompt to View Persistence Session

```
show lb persistentSessions myVserver
```

Clearing of Persistence Sessions

You might need to clear persistence sessions from the NetScaler if sessions fail to time out. You can do one of the following:

- Clear all sessions for all virtual servers at once.
- Clear all sessions for a given virtual server at once.

To Clear Persistence Sessions Using Configuration Utility

1. In the navigation pane, click **Load Balancing**.
2. In the details pane, under **Monitor Sessions**, click **Clear persistence sessions**.
3. In the **Clear Persistence Sessions** dialog box, do one of the following:
   - If you want to clear all sessions for all virtual servers on the system, in `Virtual Server`, select `All Virtual Servers`.
   - If you want to clear all sessions for a given virtual server, in `Virtual Server`, select the virtual server.
4. Click **OK**.

© Copyright 2012 Citrix Systems, Inc.

Persistence Session Clearing Using Command-line Interface

At the NetScaler command line, type the following commands to clear persistence sessions and verify the configuration:

```
clear lb persistentSessions vServer
```

```
show persistentSessions
```

Example of NetScaler Command Prompt to Clear Persistence Sessions

```
clear persistentSessions lbvip1
```

```
show persistentSessions
```

 © Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

1. You have services bound to a virtual server that uses a custom load-balancing server. If you do not bind a load monitor to the services, what will happen to the traffic that is sent to that virtual server?
    a. Traffic will be dropped.
    b. The server will respond with a "Not Found" error.
    c. The server will use the round robin load balancing method.
    d. The server will use the default least connections load balancing method.
2. Why would you want to configure persistence on a load-balancing virtual server that handles certain types of web applications, such as shopping cart applications?
    a. It is necessary to ensure the load-balancing virtual server calculates least response time correctly.
    b. It is necessary to ensure the maintenance of state information about client connections is not interrupted.
    c. It is necessary to ensure load balancing of outbound traffic across multiple Internet connections provided by different server providers.
3. If most client requests flow through a NAT device, which persistence method would not be the most efficient persistence method?
    a. Cookie-insert persistence
    b. Source-IP-address persistence
    c. SSL Session ID persistence
    d. URL Passive Persistence
4. Which two types of persistence can you configure for persistence groups? (Choose two.)
    a. Destination IP address persistence
    b. HTTP cookie-based persistence
    c. Rule-based persistence
    d. Source-IP-based persistence
    e. URL passive persistence

© Copyright 2012 Citrix Systems, Inc.

Load-Balancing Configuration Protection

When a load-balancing virtual server fails, or when the virtual server is unable to handle excessive traffic, the load-balancing setup can fail. You can protect your load-balancing setup against failure by configuring the NetScaler system to redirect excess traffic to an alternate URL, configuring a backup load-balancing virtual server, and configuring stateful connection failover.

Backup Load-Balancing Virtual Server Configuration

You can configure the NetScaler system to direct requests to a backup virtual server in the event that the primary load-balancing virtual server is DOWN or unavailable. The backup virtual server is a proxy and is transparent to the client. The system can also send a notification message to the client regarding the site outage.

You can configure a backup-load balancing virtual server when you create it, or you can change the optional parameters of an existing virtual server. You can also configure a backup virtual server for an existing backup virtual server, thus creating cascading backup virtual servers.

For more information about configuring backup load-balancing virtual servers, see Citrix article CTX132359 at *http://support.citrix.com*.

Diverting Excess Traffic to a Backup Virtual Server

In addition to taking over for a primary virtual server when it becomes unavailable, a backup load-balancing virtual server can handle excess traffic when the primary virtual server reaches its limit. To set this up, you configure the spillover option to divert new connections to the backup virtual server when the number of connections to the primary virtual server exceeds the threshold. You can allow the NetScaler system to calculate the threshold dynamically or you can configure the value manually.

You can configure persistence with spillover. When persistence is configured, connections that are diverted to the backup virtual server are not moved back to the primary virtual server after the number of connections on it drops below the threshold. Instead, the primary virtual server accepts new client connections.

For more information about diverting excess traffic to a backup virtual server, see Citrix article CTX132359 at *http://support.citrix.com*.

Spillover Configuration

There are three typs of spillover that can be configured:

- Connection-based spillover
- Dynamic spillover
- Bandwidth-based spillover

 © Copyright 2012 Citrix Systems, Inc.

Connection-Based Spillover

You can use connection-based spillover to configure a maximum threshold for the number of active client connections on a virtual server. When the client connections exceed the configured threshold limit, new client connections are diverted to the backup virtual server.

Dynamic Spillover

When you configure dynamic spillover, if the number of client connections to the primary load-balancing virtual server exceeds the sum of the maximum client values, new connections are diverted to the backup virtual server.

To configure dynamic spillover, you must first enable it on the primary virtual server. Next, you configure each service that is bound to that virtual server with appropriate maximum client values for that service. Different services can have different maximum client values. If the value for maximum client is set to 0, the spillover limit is treated as infinity, and spillover never occurs.

Content-based virtual servers do not support dynamic spillover.

Bandwidth-based Spillover

With bandwidth-based spillover, when the bandwidth used by the primary load-balancing virtual server exceeds the specified bandwidth threshold value, the NetScaler system diverts new connections to the backup virtual server. You can also configure the backup virtual server with a threshold value. When the threshold for the backup virtual server is reached, the system diverts new client connections to the next backup virtual server.

Failover Configuration

You can set up connection failover in either stateless or stateful mode. In the stateless connection failover mode, the high availability nodes do not exchange any information about the connections that are failed over. This method has no runtime overhead. In the stateful connection failover mode, the primary system synchronizes the data of the failed-over connections with the new secondary system.

For more information about stateless and stateful failover, see Citrix article CTX132359 at *http://support.citrix.com*.

Test Your Knowledge

1. Which three ways can you protect your load-balancing virtual server against failure? (Choose three.)
    a. Configure stateful connection failover.
    b. Configure a backup load-balancing virtual server.
    c. Configure the NetScaler to redirect excess traffic to an alternate URL.
    d. Configure a virtual server group and then configure link load balancing.
    e. Configure a virtual server group and then configure persistence for the entire group.
2. True or False? When you configure persistence with spillover, connections that are diverted to the backup virtual server are moved back to the primary virtual server after the number of connections on it drops below the threshold.
    a. True
    b. False

 © Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

Match the following connecton failover modes with the correct descriptions. Terms may be used more than once.

- stateless mode
- stateful mode

Description	Term
In this mode, the high availability nodes do not exchange any information about the connections that are failed over.	
In this mode, the primary system synchronizes the data of the failed-over connections with the new secondary system.	
This method has no runtime overhead.	

© Copyright 2012 Citrix Systems, Inc.

Load-Balancing Setup Management

An existing load-balancing setup does not require a great deal of work to maintain as long as it is unchanged, but most do not remain unchanged for long. Increasing load requires new load-balanced servers and eventually new NetScaler systems, which must be configured and added to the existing setup. Old servers wear out and need to be replaced, requiring removal of some servers and addition of others. Upgrades to your networking equipment or changes to topology may also require modifications to your load-balancing setup. Therefore, you will need to perform operations on server objects, services, and virtual servers. The Visualizer can display your configuration graphically, and you can perform operations on the entities in the display. You can also take advantage of a number of other features that facilitate management of the traffic through your load-balancing setup.

Services Management

Services are enabled by default when you create them. You can disable or enable each service individually. When disabling a service, you normally specify a wait time during which the service continues to handle established connections, but rejects new connections before shutting down. If you do not specify a wait time, the service shuts down immediately. During the wait time, the state of the service is OUT OF SERVICE.

You can remove a service when it is no longer used. When you remove a service, it is unbound from its virtual server and deleted from the NetScaler configuration.

For more information about managing services, see Citrix article CTX132359 at *http://support.citrix.com*.

Advanced Services Options

In addition to configuring virtual servers, you can configure advanced settings for services. The sections below cover several advanced settings.

Gradually Stepping Up the Load on a New Service with Virtual Server–Level Slow Start

You can configure the NetScaler system to gradually increase the load on a service immediately after the service is either added to a load-balancing configuration or has a state change from DOWN to UP. You can either increase the load manually with load values and intervals of your choice or configure the system to increase the load at a specified interval until the service is receiving as many requests as the other services in the configuration. During the ramp-up period for the new service, the system uses the configured load-balancing method.

This functionality is not available globally. It has to be configured for each virtual server. The functionality is available only for virtual servers that use one of the following load-balancing methods:

 © Copyright 2012 Citrix Systems, Inc.

- Round robin
- Least connection
- Least response time
- Least bandwidth
- Least packets
- Least Response Time
- Custom load

The No-Monitor Option for Services

If you use an external system to perform health checks on the services and do not want the NetScaler system to monitor the health of a service, you can set the no-monitor option for the service. If you do so, the system does not send probes to check the health of the service but shows the service as UP. Even if the service goes DOWN, the system continues to send traffic from the client to the service as specified by the load-balancing method.

The monitor can be in the ENABLED or DISABLED state when you set the no-monitor option, and when you remove the no-monitor option, the earlier state of the monitor is resumed.

You can set the no-monitor option for a service when creating the service. You can also set the no-monitor option on an existing service.

Protecting Applications on Protected Servers Against Traffic Surges

The NetScaler provides the surge protection option to maintain the capacity of a server or cache. The NetScaler regulates the flow of client requests to servers and controls the number of clients that can simultaneously access the servers. The NetScaler blocks any surges passed to the server, thereby preventing overloading of the server. For surge protection to function correctly, you must enable it globally.

Directing Requests to a Custom Web Page

The NetScaler provides the SureConnect option to ensure the response from an application. For SureConnect to function correctly, you must set it globally.

Enabling Access to Services When Down

You can enable access to a service when it is disabled or in a DOWN state by configuring the NetScaler system to use layer-2 mode to bridge the packets sent to the service. Normally, when requests are forwarded to services that are DOWN, the request packets are dropped. When you enable the Access Down setting, however, these request packets are sent directly to the load balanced servers.

Setting a Limit on the Number of Client Connections

You can specify a maximum number of client connections that each load balanced server can handle. The NetScaler system then opens client connections to a server only until this limit is reached. When the load balanced server reaches its limit, monitor probes are skipped, and the server is not used for load balancing until it has finished processing existing connections and frees up capacity.

Setting a Limit on Number of Requests for each Connection to the Server

The NetScaler system can be configured to reuse connections to improve performance. In some scenarios, however, load balanced web servers may have issues when connections are reused for too many requests. For HTTP or SSL services, use the max request option to limit the number of requests sent through a single connection to a load balanced web server.

You can configure the max request option for HTTP or SSL services only.

Setting a Threshold Value for the Monitors Bound to a Service

The NetScaler system designates a service as UP only when the sum of the weights of all monitors bound to it and that are UP is equal to or greater than the threshold value configured on the service. The weight for a monitor specifies how much that monitor contributes to designating the service to which it is bound as UP.

Setting a Timeout Value for Idle Client Connections

You can configure the service with a time-out value to terminate any idle client connections when the configured time elapses. If the client is idle during the configured time, the NetScaler closes the client connection.

Setting a Timeout Value for Idle Server Connections

You can configure a service with a timeout value to terminate any idle server connections when the configured time elapses. If the server is idle for the configured amount of time, the NetScaler system closes the server connection.

Setting a Limit on the Bandwidth Usage by Clients

In some cases, servers may have limited bandwidth to handle client requests and may become overloaded. To prevent overloading a server, you can specify a maximum limit on the bandwidth

 © Copyright 2012 Citrix Systems, Inc.

processed by the server. The NetScaler system forwards requests to a load balanced server only until this limit is reached.

Redirecting Client Requests to a Cache

You can configure a service to redirect client requests to a cache, and forward only those requests that are cache misses to a service chosen by the configured load-balancing method.

Service Groups

You can change the settings of the services in a service group, and you can perform tasks such as enabling, disabling, and removing service groups. You can also unbind members from a service group.

For more information about service groups, see Citrix article CTX128670 at *http://support.citrix.com*

Modifying a Service Group

You can modify attributes of service group members. You can set several attributes of the service group, such as maximum client, SureConnect, and compression. The attributes are set on the individual servers in the service group. You cannot set parameters on the service group such as transport information (IP address and port), weight, and server ID.

A parameter you set for a service group is applied to the member servers in the group, not to individual services.

Removing a Service Group

When you remove a service group, the servers bound to the group retain their individual settings and continue to exist on the NetScaler system.

Unbinding a Member from a Service Group

When you unbind a member from the service group, the attributes set on the service group will no longer apply to the member that you unbound. The member services retains its individual settings, however, and continues to exist on the NetScaler system.

Unbind a Service Group from a Virtual Server

When you unbind a service group from a virtual server, the member services are unbound from the virtual server and continue to exist on the NetScaler system.

Unbind Monitors from a Service Group

When you unbind a monitor from a service group, the monitor that you unbound no longer monitors the individual services that constitute the group.

Enable or Disable a Service Group

When you enable a service group and the servers, the services belonging to the service group are enabled. Similarly, when a service belonging to a service group is enabled, the service group and the service are enabled. By default, service groups are enabled.

After disabling an enabled service, you can view the service using the configuration utility or the command-line interface to see the amount of time that remains before the service goes DOWN.

Load-Balancing Virtual Servers Bound to a Service Group

In large-scale deployments, the same service group can be bound to multiple load-balancing virtual servers. In such a case, instead of viewing each virtual server to see the service group it is bound to, you can view a list of all the load-balancing virtual servers bound to a service group.

Virtual Servers Management

Virtual servers are enabled by default when you create them. You can disable and reenable virtual servers manually. If you disable a virtual server, the virtual server's state appears as OUT OF SERVICE, and it cannot accept new connections. It continues to serve requests on existing connections.

You remove a virtual server only when you no longer require the virtual server. Before you remove it, you must unbind all services from it.

For more information about managing a load-balancing virtual server, see Citrix article CTX132359 at *http://support.citrix.com*.

The Load-Balancing Visualizer

The Load-Balancing Visualizer is a tool that you can use to view and modify the load-balancing configuration in graphical format. Following is an example of the Visualizer display:

You can use the visualizer to view the following:

- The services and service groups that are bound to a virtual server.
- The monitors that are bound to each service.
- The policies that are bound to the virtual server.
- The policy labels, if configured.

 © Copyright 2012 Citrix Systems, Inc.

- Configuration details of any displayed element.
- Load-balancing virtual server statistics.
- Statistical information such as the number of requests received each second by the virtual server and the number of hits each second for rewrite, responder, and cache policies.
- A comparative list of all the parameters whose values either differ or are not defined across service containers.

You can also use the Visualizer to add and bind new objects, modify existing ones, and enable or disable objects.

The Visualizer requires a graphic interface, so it is available only through the configuration utility.

For more information about the load-balancing virtualizer, see Citrix article CTX132359 at *http://support.citrix.com*

Traffic Management

Managing client connections properly helps to ensure that your applications remain available to end users even when your NetScaler system is experiencing high loads. A number of load-balancing features and other features available on the system can be integrated into a load-balancing setup to process load more efficiently, divert it when necessary, and prioritize the tasks that the system must perform.

Configuration of Sessionless Load-Balancing Virtual Servers

When the NetScaler system performs load balancing, it creates and maintains sessions between clients and servers. The maintenance of session information places a significant load on the NetScaler resources, and sessions might not be needed in scenarios such as a direct server return (DSR) setup and the load balancing of intrusion detection systems (IDS). To avoid creating sessions when they are not necessary, you can configure a virtual server on the NetScaler for sessionless load balancing. In sessionless load balancing, the NetScaler carries out load balancing on each packet.

Sessionless load balancing can operate in MAC-based forwarding mode or IP-based forwarding mode.

For more information about sessionless load balancing, see Citrix article CTX132359 at *http://support.citrix.com*

Direct Request According to Priority

The NetScaler system supports prioritization of client requests with its priority queuing feature. This feature allows you to designate certain requests, such as those from important clients, as priority requests so that the system responds to them first. This allows you to provide uninterrupted service to those clients through demand surges or DDoS attacks on your web site.

For more information about directing requests according to priority, see Citrix article CTX132359 at *http://support.citrix.com*

.

Delayed Cleanup of Virtual Server Connections Enablement

Under certain conditions, you can configure the `downStateFlush` setting to terminate existing connections when a service or a virtual server is marked DOWN. Terminating existing connections frees resources and, in certain cases, speeds recovery of overloaded load-balancing setups.

A virtual server is marked DOWN only when all services bound to it are marked DOWN. When a virtual server goes DOWN, it terminates all connections, either immediately or after allowing existing connections to complete.

 © Copyright 2012 Citrix Systems, Inc.

> You must not enable the downStateFlush setting on those application servers that must complete their transactions. You can enable this setting on web servers whose connections can safely be terminated when they marked DOWN.

For more information about enabling delayed cleanup of virtual server connections, see Citrix article CTX132359 at *http://support.citrix.com*.

Graceful Shutdown of Services

If you want to disable a service only when all the established connections are closed by the server or the client, you can use the graceful shutdown option.

During scheduled network outages such as system upgrades or hardware maintenance, you may have to close or disable some services. To avoid disrupting sessions that have already been established, you can specify a wait time, which places a service in the transition out of service (TROFS) state until the specified wait time expires. The service then enters the out of service (OFS) state.

Often, however, you cannot estimate the amount of time needed for all the connections to a service to complete the existing transactions. If a transaction is unfinished when the wait time expires, shutting down the service may result in data loss. In this case, you can specify graceful shutdown for the service, so that the service is disabled only when all the established connections are closed by either the server or the client.

Persistence is maintained according to the specified method even if you enable graceful shutdown. The system continues to serve all the persistent clients, including new connections from the clients, unless the service is marked DOWN during the graceful shutdown state as a result of the checks made by a monitor.

For more information about graceful shutdown of services, see Citrix article CTX132359 at *http://support.citrix.com*.

Use of a Specified Source IP for Back-end Communications

In many situations, you may want the NetScaler to use a specific IP address or any IP address from a specific set of IP addresses for backend communications. The following are a few examples:

- A server can distinguish monitor probes from traffic if the source IP address used for monitor probes belongs to a specific set.
- To improve server security, a server may be configured to respond to requests from a specific set of IP addresses or, sometimes, from a single specific IP address. In such a case, the NetScaler can use only the IP addresses accepted by the server as the source IP address.
- The NetScaler can manage its internal connections efficiently if it can distribute the MIPs or SNIPs into IP sets and use an address from a set only for connecting to a specific service.

To configure the NetScaler to use a specified source IP address, create net profiles (network profiles) and configure the NetScaler entities to use the profile. A net profile can be bound to load-balancing or content-switching virtual servers, services, service groups, or monitors.

If a net profile is bound to a load-balancing or content-switching virtual server, the profile will be used for sending traffic to all the services bound to it. If a net profile is bound to a service group, NetScaler uses the profile for all the members of the service group. If a net profile is bound to a monitor, NetScaler uses the profile for all the probes sent from the monitor.

For more information about using a specified source IP for backend communications, see Citrix article CTX132359 at *http://support.citirx.com*.

Enabling of TCP Buffering of Responses

The NetScaler system provides a TCP buffering option that buffers only responses from the load balanced server. This enables the system to deliver server responses to the client at the maximum speed that the client can accept them.

TCP buffering set at the service level takes precedence over the global setting.

For more information about enabling of TCP buffering of responses, see Citrix article CTX132359 at *http://support.citrix.com*.

Maintaining of Client Connection for Multiple Client Requests

You can set the client keep-alive parameter to configure an HTTP or SSL service to keep a client connection to a web site open across multiple client requests. If client keepalive is enabled, even when the load balanced web server closes a connection, the NetScaler system keeps the connection between the client and itself open. This setting allows services to serve multiple client requests on a single client connection.

If you do not enable this setting, the client will open a new connection for every request that it sends to the web site. The client keep-alive setting saves the packet round trip time required to establish and close connections. This setting also reduces the time to complete each transaction. Client keep-alive can be enabled only on HTTP or SSL service types.

Client keep-alive set at the service level takes precedence over the global client keepalive setting.

For more information about enabling of Client Keepalive, see Citrix article CTX132359 at *http://support.citrix.com*.

 © Copyright 2012 Citrix Systems, Inc.

Use of the Source IP Address of the Client When Connecting to the Server

You can configure the NetScaler system to forward packets from the client to the server without changing the source IP address. This is useful when you cannot insert the client IP address into a header, such as when working with non-HTTP services.

When this is configured you will lose most the benefit of TCP multiplexing.

For more information about configuring USIP globally, see Citrix article CTX132369 at *http://support.citrix.com*.

For more information about using the source IP address of the client when connecting to the server, see Citrix article CTX132359 at *http://support.citrix.com*.

Test Your Knowledge

Match the following terms with the correct information.

- Compression
- downStateFlush
- Client Keepalive
- TCP Buffering of Responses
- Graceful Shutdown of Services
- Direct Request According to Priority
- Sessionless Load-Balancing Virtual Servers
- Use of the Source IP Address of the Client when connecting to the Server

Information	Term
When this is configured, you will lose most of the benefit of TCP multiplexing.	
When this is enabled, even when the load-balanced web server closes a connection, NetScaler keeps the connection between the client and itself open.	
This reduces bandwidth requirements and can significantly improve server responsiveness in bandwidth-constrained setups.	
This enables the system to deliver server responses to the client at the maximum speed that the client can accept them.	
This will disable a service only when all the established connections are closed by the server or the client.	
You must not enable this setting on application servers that must complete their transactions.	
This feature allows you to designate certain requests as priority requests which ensures the system responds to them first.	

© Copyright 2012 Citrix Systems, Inc.

Information	Term
You would configure this in scenarios such as a direct server return (DSR) setup and the load blancing of intrusion detection systems (IDS).	

© Copyright 2012 Citrix Systems, Inc.

Module 8

Global Server Load Balancing

© Copyright 2012 Citrix Systems, Inc.

Overview

Global Server Load Balancing (GSLB) directs DNS requests to the best-performing GSLB site in a distributed Internet environment. GSLB enables distribution of traffic across multiple sites, manages disaster recovery, and ensures that applications are consistently accessible.

After completing this module, you will be able to:

- Identify GSLB architecture, basic concepts, and common deployment methods.
- Configure GSLB on the NetScaler system.
- Configure Metric Exchange Protocol (MEP) to check the health of datacenters, remote servers, and ensure availability.
- Configure site-to-site communication to synchronize a primary GSLB site with secondary sites and exchange site metrics.
- Customize the GSLB configuration.
- Monitor the GSLB configuration.
- Configure GSLB to protect against failure.

© Copyright 2012 Citrix Systems, Inc.

GSLB Deployment Methods

NetScaler systems configured for GSLB provide for disaster recovery and ensure that applications are continuously available by protecting against points of failure in a wide area network (WAN). GSLB can balance the load across datacenters by directing client requests to the closest or best-performing datacenter or to surviving datacenters in case of an outage. The following are typical GSLB configurations:

Active-active datacenter setup	Consists of multiple active datacenters. Client requests are load balanced across active datacenters.
Active-standby datacenter setup	Consists of an active and a standby datacenter. When a failover occurs as a result of a disaster event, the standby datacenter becomes operational.
Proximity setup	Directs client requests to the datacenter that is closest in geographical distance or network distance.

In a typical configuration, a local DNS server sends client requests to a GSLB virtual server where the GSLB services are bound. A GSLB service identifies a load-balancing or content-switching virtual server, which can be at the local site or a remote site. If the GSLB virtual server selects a load-balancing or content-switching virtual server at a remote site, it sends the virtual server's IP address to the DNS server, which sends it to the client. The client then re-sends the request to the new virtual server at the new IP.

The GSLB entities that you must configure are the GSLB sites, the GSLB services, the GSLB virtual servers, load-balancing or content-switching virtual servers, and authoritative DNS (ADNS) services. You must also configure MEP. You can configure DNS views to expose different parts of your network to clients accessing the network from different locations.

 © Copyright 2012 Citrix Systems, Inc.

GSLB Concepts

GSLB is a DNS-based solution that load balances services between geographically distributed locations. GSLB operates under many of the same general principles as load balancing, but it relies on DNS for directing client requests.

With ordinary DNS, when a client sends a DNS request, it receives a list of IP addresses of the domain or service. Generally, the client chooses the first IP address in the list and initiates a connection with that server. The DNS server uses a technique called DNS round robin to rotate through the IPs on the list, sending the first IP address to the end of the list and promoting the others after it responds to each DNS request. This technique ensures equal distribution of the load, but it does not support disaster recovery, load balancing based on load or proximity of servers, or persistence.

When you configure GSLB and enable MEP, the NetScaler systems use the DNS infrastructure to connect the client to the datacenter that best meets the criteria that you set. The criteria can designate the least-loaded datacenter, the closest datacenter, the datacenter that responds most quickly to requests from the client's location, a combination of those metrics, and SNMP metrics. An appliance keeps track of the location, performance, load, and availability of each datacenter and uses these factors to select the datacenter to which to send a client request.

A GSLB configuration consists of a group of GSLB entities on each appliance in the configuration. These entities include GSLB sites, GSLB services, GSLB virtual servers, load-balancing or content-switching servers, and ADNS services.

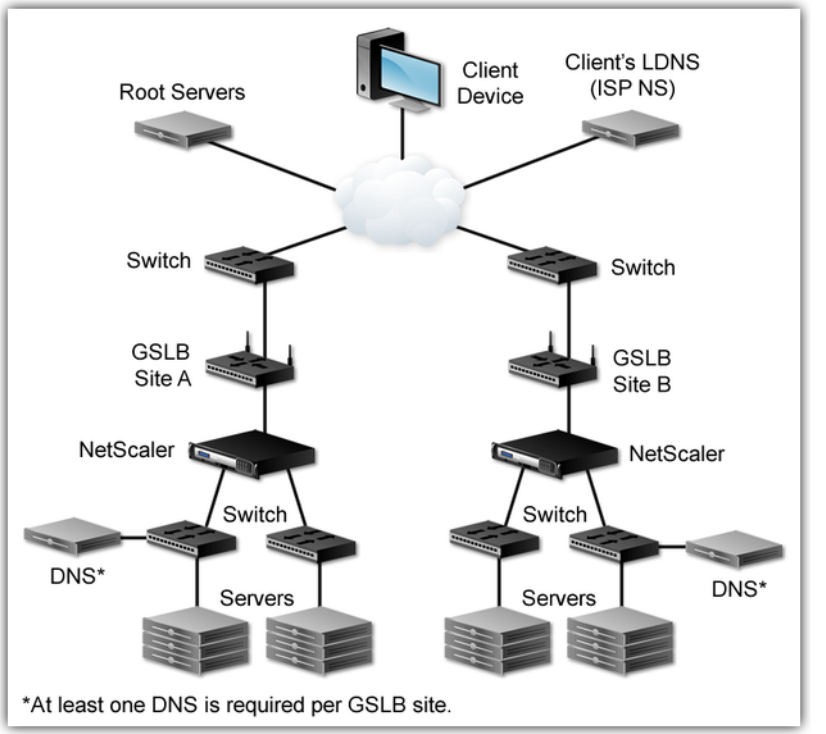
Typical uses of GSLB include:

- Network traffic distribution across multiple sites
- Server load distribution across multiple sites
- Disaster recovery
- Protection against points of failure in a wide area network (WAN)

GSLB Hierarchy

GSLB networks have a maximum of 32 sites. NetScaler systems can be deployed in hierarchy mode for scenarios that require additional units.

GSLB Conversation Process



The figure and the following steps illustrate a typical GSLB conversation process. The asterisks in the figure denote that the back-end DNS server is necessary in a proxy DNS configuration only. The NetScaler system answers the site DNS requests in ADNS configurations.

1. The end user enters www.example.com in a web browser.
2. The client sends a DNS look-up query for www.example.com to the name server that is configured.
3. The name server returns the IP address for a name server that is authoritative for www.example.com as delivered by the root server. The returned address is the one registered for the www.siteA.example.com site. The top-level servers (root servers) circle through the list round robin and return the next IP address in line.
4. The client queries the NetScaler system in the GSLB configuration at the IP address returned in the prior step. The NetScaler system, based on its configured load-balancing method, returns the IP address the client needs to query for the service for which it is looking, such as HTTP and HTTPS.
5. The responding NetScaler system queries the back-end DNS server for the address to serve the look-up request if the GSLB configuration is a proxy DNS configuration.

 © Copyright 2012 Citrix Systems, Inc.

The site to which the NetScaler system directs the client is either:

- A site the NetScaler system is hosting within the load-balancing configuration
- Another GSLB site within the membership of sites

GSLB Entities

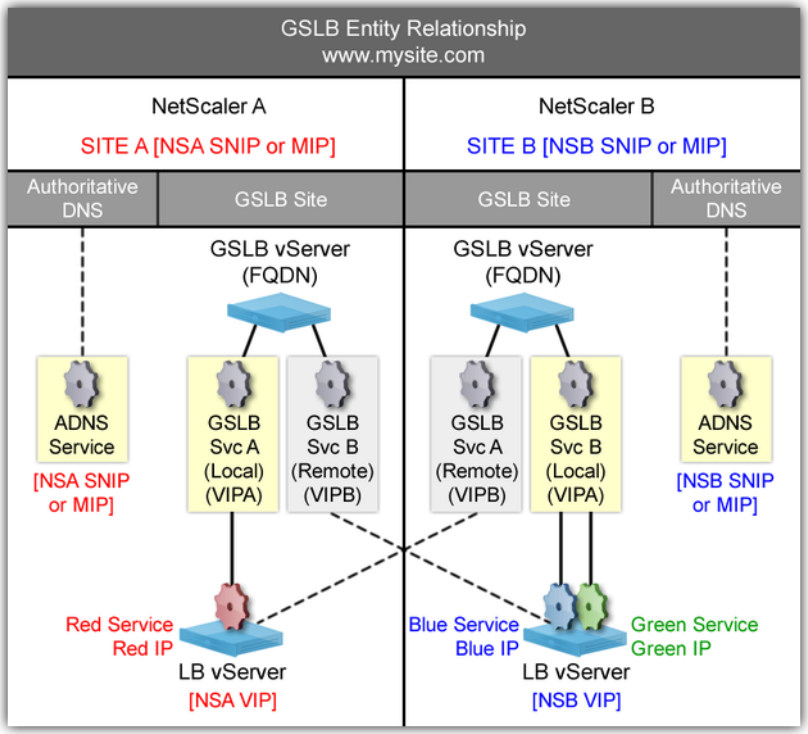
A GSLB configuration includes entities on the NetScaler system that direct client traffic to applications and resources. The following items are entities in a GSLB environment:

GSLB site	A GSLB site is typically a datacenter in which a NetScaler system is located. The terms local site and remote site are used to refer to the site in relation to the NetScaler systems in the GSLB deployment. Each GSLB site is managed by a NetScaler system that is local to that site. Each of these systems treats its own site as the local site and all other sites, managed by other systems, as remote sites.
GSLB service	A GSLB service is a representation of a load-balancing or content-switching virtual server, although it can represent any type of virtual server. The GSLB service determines how incoming traffic is routed.
GSLB virtual server	A GSLB virtual server enables client requests to be forwarded to the appropriate GSLB site. A GSLB virtual server is assigned one or more GSLB services, and load balances the incoming traffic among the services. The GSLB virtual server evaluates the configured GSLB methods (algorithms) to select the appropriate service to which to send a client request. DNS virtual servers are only necessary in a DNS proxy configuration. Otherwise, in an ADNS configuration, each GSLB site will use the locally configured DNS service with mirrored static DNS records for each site in the configuration.
Load-balancing or content-switching virtual server	Load-balancing or content-switching virtual servers load balance incoming traffic to the appropriate server.
ADNS service	The ADNS service accepts incoming client requests for domains for which the NetScaler system is authoritative.

GSLB Entity Relationship

DNS virtual servers are only necessary in a DNS proxy configuration. Otherwise, in an ADNS configuration, each GSLB site will use a locally configured DNS service with mirrored static DNS records for each site in the configuration.

 © Copyright 2012 Citrix Systems, Inc.



A GSLB virtual server has one or more GSLB services bound to it, and load balances traffic among those services. It evaluates the configured GSLB methods to select the appropriate service to which to send a client request. Because the GSLB services can represent either local or remote servers, selecting the optimal GSLB service for a request has the effect of selecting the datacenter that should serve the client request.

The domain for which GSLB is configured must be bound to the GSLB virtual server, because one or more services bound to the virtual server will serve requests made for that domain.

Unlike other virtual servers configured on a NetScaler system, a GSLB virtual server does not have its own virtual IP address (VIP).

GSLB DNS Methods

You can configure the NetScaler system to respond to DNS queries in the following ways:

- ADNS
- DNS proxy
- DNS sub-delegation

The NetScaler system in ADNS mode is the authoritative source for a domain and all the records within that domain. It resolves the domain name received in the client DNS query to one or more IP addresses. It does not support zone transfers. This method:

- Occurs when the NetScaler system is assigned and is authoritative for a domain
- Is typically used for GSLB internal-facing deployments
- Presents challenges with Active Directory

For the NetScaler to be authoritative, you must also create start of authority (SOA) and name server (NS) records. For more information about SOA and NS records, see Citrix eDocs at *http://edocs.citrix.com*.

As a proxy to an external DNS server, the NetScaler system caches the DNS responses sent by the external name server. This method:

- Occurs when the NetScaler system forwards or proxies DNS requests to a DNS server
- Does not have an internal DNS on the NetScaler system

DNS sub-delegation occurs when the DNS server assigns the responsibility for a subdomain to a NetScaler system. The DNS server redirects DNS requests to the NetScaler system that resolves the IP address. This method:

- Occurs when the DNS server assigns the responsibility for a subdomain to a NetScaler system
- Has the DNS server redirecting DNS requests to the NetScaler that resolves the IP address

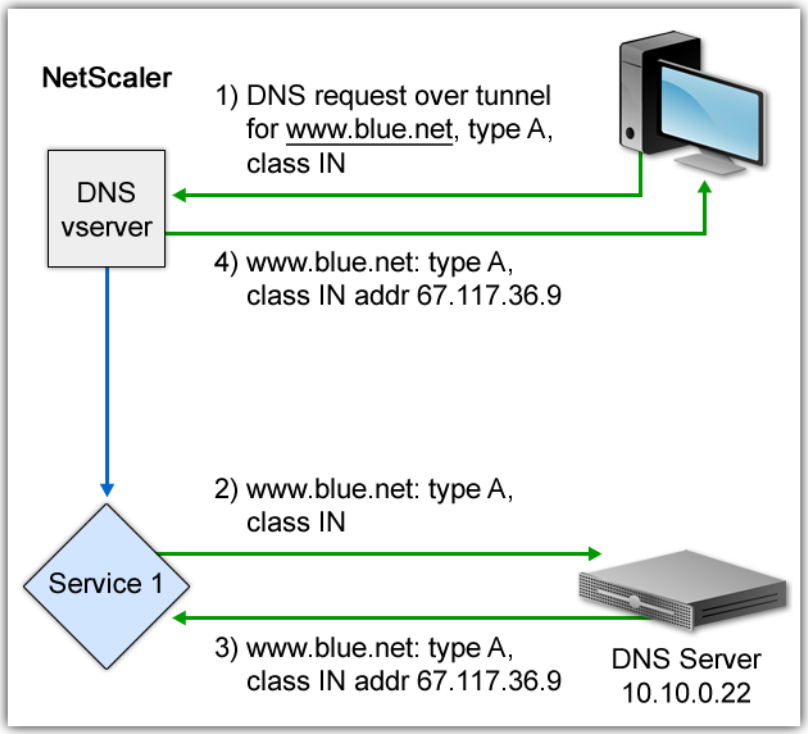
Name Servers

Name servers store information about one or more zones. The following list contains descriptions of name servers that you can add internally on the NetScaler system:

IP address-based	The end user has to specify the IP address of the name server to contact.
Virtual server-based	The end user has to specify the name of the DNS virtual server configured in the NetScaler system.

 © Copyright 2012 Citrix Systems, Inc.

External DNS Server Example

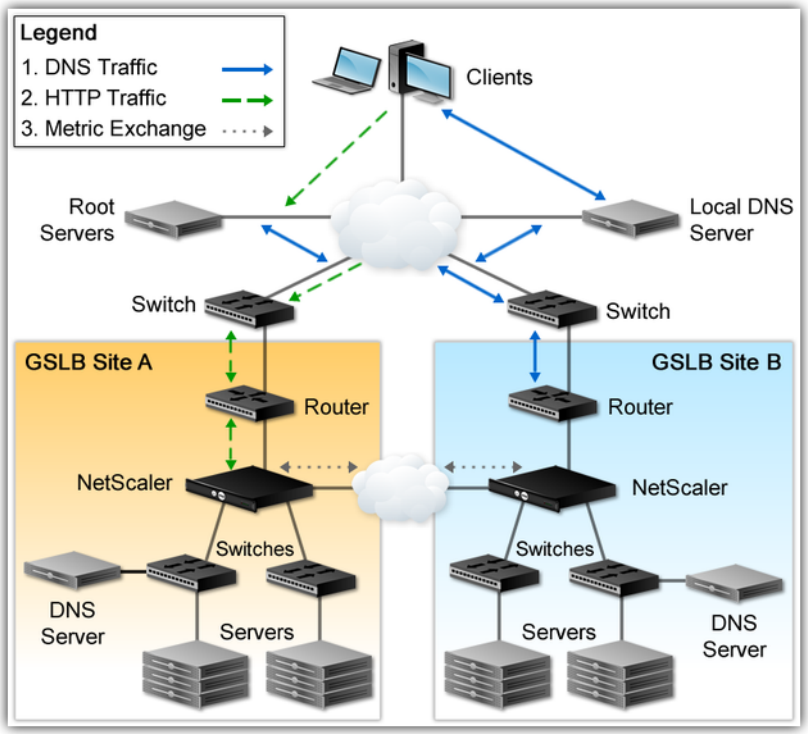


The figure illustrates an example of an external DNS server with a VPN connection:

1. The end user opens a web browser locally and enters www.blue.net into the browser. Split tunneling is disabled. The VPN client intercepts the DNS query, packages it in an HTTP request, and forwards it over the tunnel.
2. A load-balancing decision is made, and the request is sent to a back-end DNS server for resolution.
3. The DNS server hands the response back to the NetScaler system.
4. The DNS virtual server wraps the response in HTTP and forwards it across the tunnel back to the client. The VPN client unwraps the HTTP request and forwards the DNS response to the client.

© Copyright 2012 Citrix Systems, Inc.

DNS Proxy Configuration



The figure illustrates a DNS proxy configuration. In a proxy configuration, the NetScaler system passes domain requests to the DNS server. The following considerations can apply in a proxy configuration:

- If the NetScaler system is authoritative for the requested zone, the NetScaler system responds to the query.
- If the request is for a zone that is within the GSLB domain, the NetScaler system responds with the optimal virtual IP server address in the GSLB domain.

DNS Request Scenarios

The NetScaler system uses existing virtual server and service functionality when responding to DNS requests. The NetScaler system uses a MIP address or SNIP address as the source IP address for queries. The DNS server responses are cached on the NetScaler system until the time to live (TTL) expires.

For clients making DNS requests, two different scenarios exist:

1. Local DNS

 © Copyright 2012 Citrix Systems, Inc.

2. Load-balancing DNS

In a local DNS scenario:

- A local DNS server is created on the NetScaler system.
- The local DNS is an ADNS server for the zone configured.
- The local DNS listens on an IP address provided in the configuration. Clients can configure their local TCP/IP stack to forward queries to this IP address.

In a load-balancing DNS scenario:

- A DNS load-balancing virtual server is created and provides an IP address.
- The services are added to redirect traffic to DNS servers.
- The clients configure the load-balancing virtual server IP address as their DNS server IP address.

Resource Records

On the NetScaler system, DNS records contain DNS data. The following table lists the record types and the number of records for each record type that you can configure for a domain on the NetScaler:

Record Type	Number of Records
Address (A)	25
IPv6 (AAAA)	5
Mail exchange (MX)	12
Name server (NS)	16
Service (SRV)	8
Pointer (PTR)	20
Canonical name (CNAME)	1
Start of Authority (SOA)	1
Text (TXT)	20

Authoritative DNS Service

You can configure the NetScaler system with single or with multiple instances of an ADNS server, where each instance listens on a different IP address, and all instances reference the same name table. The ADNS service is a local service type listening for incoming DNS requests on UDP port 53.

In an authoritative configuration, the NetScaler system answers the DNS query. In this configuration, the NetScaler system:

- Is locally configured as SOA for the GSLB domain
- Can be configured for a maximum of 32 sites
- Does not support zone transfers or recursive query
- Can be set to participate as authoritative

 © Copyright 2012 Citrix Systems, Inc.

Viewing and Configuring GSLB with the Visualizer

The configuration utility includes a GSLB visualizer tool, which provides an alternative way to view and configure entities in a GSLB configuration. The visualizer displays all configured GSLB domains, GSLB services, GSLB sites, ADNS services, and any monitors that are bound to the services. It also displays all the load-balancing, content-switching, cache-redirection, and Access Gateway virtual servers that the GSLB services represent.

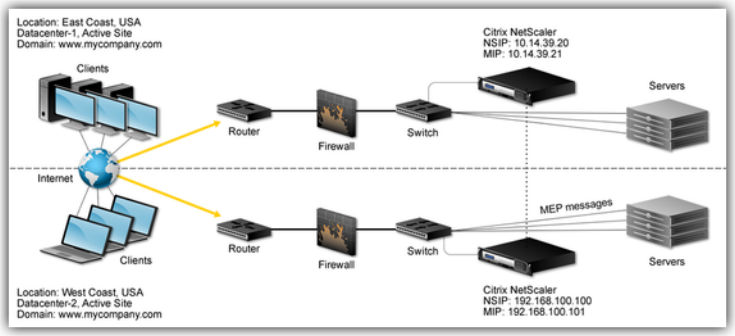
If you want to view the configurations of remote GSLB sites, you must configure the sites with public IP addresses and enable management access for each of them.

You can use the GSLB visualizer to perform the following GSLB configuration tasks:

- Add, view, and configure GSLB domains and GSLB services
- View and configure GSLB sites and ADNS services for each site
- View and configure any monitors that are bound to the services
- View and configure the content-switching, load-balancing, cache-redirection, or Access Gateway virtual server that each GSLB service represents
- View statistics for GSLB domains, sites, ADNS services, and virtual servers
- View configuration details of any displayed entity
- View load-balancing and content-switching virtual servers
- View bindings for GSLB services, ADNS services, monitors, and virtual servers
- Enable and disable GSLB services, ADNS services, monitors, and virtual servers
- Remove a domain from the GSLB setup
- Save the visual representation of the GSLB setup as an image

© Copyright 2012 Citrix Systems, Inc.

Implementing Static GSLB



When the DNS request from the resolver of the client is received by the NetScaler system, the load-balancing and the site fault-tolerance decision is made based on the health status and the load of the participating sites. When the host name of the URL is resolved, all traffic from the client is sent directly to the resolved site.

When the DNS request from the resolver of the client is received by the NetScaler system, the site load information is exchanged between the GSLB sites. When the host name of the URL is resolved, all traffic from the client is sent directly to the resolved site. For GSLB to work as defined, either the MEP should be enabled, or explicit monitors should be bound to remote services.

MEP Failover

If the MEP fails for any of the participating sites because of external factors, then the default method of round robin is used instead of least-response time, least connections, least bandwidth, and least packets. If the remote service belonging to the site for which MEP has failed has an explicit monitor bound to it, and its state is UP, then it is included in the round-robin rotation; otherwise, it is not.

ADNS Service Configuration

When you configure the NetScaler system as an ADNS server, it accepts DNS requests from the client and responds with the IP address of the datacenter to which the client should send requests.

To create an ADNS service using the configuration utility, go to **Configuration** > **Load Balancing** > **Services** and click **Add** to create the new service.

If you are using the command-line interface, enter:

```
add service name IPAddress ADNS port
```

 © Copyright 2012 Citrix Systems, Inc.

Basic GSLB Site Configuration

A GSLB site is a representation of a datacenter in your network and is a logical grouping of GSLB virtual servers, services, and other network entities. Typically, in a GSLB setup, there are many GSLB sites that are equipped to serve the same content to a client. These are usually geographically separated to make sure that the domain is active even if one site goes DOWN completely. All of the sites in the GSLB configuration must be configured on every NetScaler system hosting a GSLB site. In other words, at each site you configure the local GSLB site and remote GSLB site.

Once GSLB sites are created for a domain, the NetScaler system sends client requests to the appropriate GSLB site as determined by the GSLB algorithms configured.

To create a GSLB site using the configuration utility, go to **Configuration** > **GSLB** > **Sites** and click **Add**.

Using the command-line interface, enter:

```
add gslb site siteName siteIPAddress
```

GSLB Service Configuration

A GSLB service is a representation of a load-balancing or content-switching virtual server. A local GSLB service represents a local load-balancing or content-switching virtual server. A remote GSLB service represents a load-balancing or content-switching virtual server configured at one of the other sites in the GSLB setup. At each site in the GSLB setup, you can create one local GSLB service and any number of remote GSLB services.

To create a GSLB service using the configuration utility, go to **Configuration** > **GSLB** > **Services** and click **Add**.

Using the configuration utility, enter:

```
add gslb service serviceName serverName | server IP serviceType
port -siteName string
```

Services Management

Services are enabled by default when you create them. You can disable or enable each service individually. When disabling a service, you normally specify a wait time during which the service continues to handle established connections, but rejects new connections before shutting down. If you do not specify a wait time, the service shuts down immediately. During the wait time, the state of the service is OUT OF SERVICE.

You can remove a service when it is no longer used. When you remove a service, it is unbound from its virtual server and deleted from the NetScaler configuration.

For more information about managing services, see Citrix article CTX132359 at *http://support.citrix.com*.

GSLB Virtual Server Configuration

A GSLB virtual server is an entity that represents one or more GSLB services and balances traffic between them. It evaluates the configured GSLB methods or algorithms to select a GSLB service to which to send the client request.

To configure a GSLB virtual server in the configuration utility, go to **Configuration** > **GSLB** > **Virtual Servers** and click **Add**.

Using the command-line interface, enter:

```
add gslb vserver name serviceType -ipType (IPv4 | IPv6)
```

Binding of GSLB Services to a GSLB Virtual Server

Once the GSLB services and virtual server are configured, relevant GSLB services must be bound to the GSLB virtual server to activate the configuration.

To bind a GSLB service to a GSLB virtual server using the configuration utility, go to **Configuration** > **GSLB** > **Virtual Servers** and select the GSLB virtual server to which you want to bind the services. Once selected, open the virtual server and select the GSLB services that you want to bind to the virtual server.

Using the command-line interface, enter:

```
bind gslb vserver name -serviceName string
```

Binding of a Domain to a GSLB Virtual Server

To make a NetScaler system the ADNS server for a domain, you must bind the domain to the GSLB virtual server. When you bind a domain to a GSLB virtual server, the NetScaler system adds an address record for the domain that contains the name of the GSLB virtual server. The SOA and NS records for the GSLB domain must be added manually.

To bind a domain to a GSLB virtual server using the configuration utility, go to **Configuration** > **GSLB** > **Virtual Servers** and select the GSLB virtual server to which you want to bind the domain, then click **Open**, select the **Domains** tab, and click **Add**.

Using the command-line interface, enter:

```
bind gslb vserver name -domainName string
```

 © Copyright 2012 Citrix Systems, Inc.

Synchronizing a GSLB Configuration

For GSLB to work, all participating NetScaler systems must share identical GSLB configurations. Updating configurations in multiple locations can be expansive and cause downtime. The auto-sync feature allows one NetScaler system to push its configurations to other units in its GSLB network.

Auto-sync:

- Aids in configuring GSLB in multiple locations
- Requires configurations only on one unit
- Overrides any GSLB configurations on the target units

Caution is highly recommended when using this command, as it will propagate a number of changes automatically. The [-preview] option allows you to view the commands that will be executed on the other NetScaler systems.

Automatically synchronize and verify a GSLB configuration by entering the following commands into the command-line interface:

```
sync gslb config -preview -forceSync string  -nowarn -debug
```

```
show gslb syncstatus
```

GSLB with Sub-Domains

When implementing GSLB, there is a 32-site maximum for each NetScaler implementation. This does not mean that the NetScaler system can only be authoritative or support 32 DNS records. The maximum is only based on the number of sites.

However, using the sub-domain delegation property of standard DNS, you can create a special sub-domain that is used explicitly for performing GSLB tasks. The NetScaler system is the ADNS server for this sub-domain. With NetScaler, you can configure parent and child GSLB sites.

For more information about configuring a GSLB sub-domain, see Citrix article CTX110488 at *http://support.citrix.com*.

Metric Exchange Protocol

The NetScaler system uses the proprietary metric exchange protocol (MEP) to exchange load metrics, network metrics, and persistence information between sites. A single, long-lived connection between two sites is used to exchange all metric information. MEP is required for health checking of datacenters to ensure their availability. A connection for exchanging network metrics can be initiated by either of the datacenters involved in the exchange, but a connection for exchanging site metrics is always initiated by the datacenter with the lower IP address. By default, this connection is made from the NSIP address to the GSLB site IP address. However, you can configure the NetScaler system to use an IP address other than the NSIP address.

Performing user authentication before metric information is exchanged provides controlled access. All sites taking part in metric exchange should have the same nsroot user ID and password.

The communication process is accomplished between each GSLB site on TCP port 3011 or port 3009 if secure. You must open the correct port on any firewalls between the NetScaler systems. The public IP address of the site needs to be allowed on any blocking firewall.

When MEP is disabled, GSLB methods are limited to round-robin, static-proximity, and source-IP-address hash. All other methods revert to round robin when MEP is disabled.

If MEP is disabled and there is no explicit monitor bound to the GSLB service, then the GSLB site will be marked as DOWN.

Metric Information Types

The GSLB site metric exchange interval is one second. Metric information types include:

Load	Load information consists of the current number of connections and current packet rate for a load-balancing virtual server.
Network	GSLB sites exchange round trip time (RTT) information about the learned DNS of the clients when dynamic proximity-based GSLB is enabled. This information is exchanged every five seconds.
Persistence	GSLB site information is exchanged every five seconds.

To Configure Network Metric Exchange

You can enable or disable the exchange of RTT information about the client's local DNS when the GSLB dynamic method RTT is enabled. This information is exchanged every five seconds.

 © Copyright 2012 Citrix Systems, Inc.

To enable or disable network metric information exchange using the command-line interface, enter:

```
set gslb site GSLBSiteName -nwmetrixExchange (ENABLED | DISABLED)
```

To enable or disable network metric information exchange using the configuration utility, follow these steps:

1. In the navigation pane, expand **Configuration** > **GSLB** > **Sites**.
2. In the details pane, select the site, and then click **Open**.
3. In the **Configure GSLB Site** dialog box, select or clear the check box next to **Network Metrics Exchange** and click **OK**.

To Configure Site Metric Exchange

Site metrics exchanged between the GSLB sites include the status of each load-balancing and content-switching virtual server, the current number of connections, the current packet rate, and current bandwidth usage information.

The NetScaler system needs this information to perform load balancing between the sites. The site metric exchange interval is one second. A remote GSLB service must be bound to a local GSLB virtual server to enable the exchange of site metrics with the remote service.

To enable or disable site metric exchange using the command-line interface, enter:

```
set gslb site GSLBSiteName -metricExchange(Enabled | Disabled)
```

When using the configuration utility, perform the following steps:

1. In the navigation pane, expand **Configuration** > **GSLB** > **Sites**.
2. In the details pane, select the site, and then click **Open**.
3. In the **Configure GSLB Site** dialog box, select or clear the check box next to **Metrics Exchange** and click **OK**.

To Configure Persistence Information Exchange

You can enable or disable the exchange of persistence information at each site. This information is exchanged every five seconds between NetScaler systems participating in GSLB.

To enable or disable persistence information exchange using the command-line interface, enter:

```
set gslb site GSLBSiteName -sessionExchange (ENABLED | DISABLED)
```

When configuring persistence information exchange using the configuration utility, perform the following steps:

1. In the navigation pane, expand **Configuration** > **GSLB** > **Sites**.

2. In the details pane, select the site, and then click **Open**.
3. In the **Configure GSLB Site** dialog box, select or clear the check box next to **Persistence Session Entry Exchange** and click **OK**.

GSLB Monitoring Configuration

The following table illustrates how MEP behaves with explicit monitors.

Monitoring	MEP Enabled	MEP Disabled
Explicit monitors	Health status controlled by monitoring	Health status controlled by monitoring
No explicit monitors	Health status controlled by MEP	All services belonging to that site are marked DOWN

MEP is used to exchange all statistics, including service health state, related to a GSLB service. If an explicit monitor is bound, the NetScaler system ignores the GSLB service state collected through MEP, and GSLB uses the state reported by the monitor.

When configuring GSLB monitoring, you can use NetScaler monitors instead of, or in addition to, MEP.

 © Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

1. Which GSLB entity is a representation of a load-balancing or content-switching virtual server?
    a. Site
    b. Service
    c. ADNS
    d. Datacenter
2. Which step must you complete to ensure that a NetScaler system is the ADNS server for a domain?
    a. Bind the domain to the GSLB virtual server.
    b. Bind the GSLB services to the GSLB virtual server.
    c. Synchronize the GSLB configuration.
    d. Enable MEP for the GSLB sites.
3. Which three GSLB methods are available if you disable MEP? (Choose three.)
    a. Round-robin
    b. Static-proximity
    c. Soure-IP-address hash
    d. Round Trip Time (RTT)
    e. Least Bandwidth
4. Which two metrics exchange GSLB-site information every five seconds? (Choose two.)
    a. Network
    b. Persistence
    c. Server
    d. Site
5. Which traditional load-balancing method does not require MEP to be enabled if explicit monitoring is configured?
    a. Source-IP-address hash
    b. Least connections
    c. Least packets
    d. Proximity-based

© Copyright 2012 Citrix Systems, Inc.

Configuring Site-to-Site Communication

GSLB site-to-site communication is between the remote procedure call (RPC) nodes that are associated with the communicating sites. A primary GSLB site establishes connections with secondary sites to synchronize GSLB configuration information and to exchange site metrics.

An RPC node is created automatically when you create a GSLB site, and is assigned an internally generated user name and password. The NetScaler system uses this user name and password to authenticate itself to remote GSLB sites during connection establishment. No configuration steps are necessary for an RPC node, but you can specify a password of your choice, enhance security by encrypting the information that GSLB sites exchange, and specify a source IP address for the RPC node.

The system needs a NetScaler-owned IP address to use as the source IP address when communicating with other GSLB sites. By default, the RPC nodes use either a subnet IP (SNIP) address or a mapped IP (MIP) address, but you can specify an IP address of your choice.

Citrix recommends that you have either a MIP or SNIP address configured in the same subnet as the NSIP address. Certain features of the NetScaler system, such as proximity-based monitors in GSLB, GSLB MEP communication or high-availability synchronization, will otherwise not function correctly.

 © Copyright 2012 Citrix Systems, Inc.

RPC Node Password

You can secure the communication between sites in your GSLB configuration by changing the password of each RPC node. After you change the password for the RPC node of the local site, you must manually propagate the change to the RPC node at each remote site.

To set the RPC node password using the configuration utility, go to **Configuration** > **Network** > **RPC** and select the RPC node for which you want to change the password.

Using the command-line interface, enter:

```
set ns rpcNode IPAddress {-password}
```

It is important to note that each password is stored in encrypted form. You can verify that the password has changed by entering the following command:

```
show rpcNode
```

Encryption of Site Metrics Exchange

You can secure the information that is exchanged between GSLB sites by setting the secure option for the RPC nodes in the GSLB setup. With the secure option set, the NetScaler system encrypts all communication sent from the node to other RPC nodes.

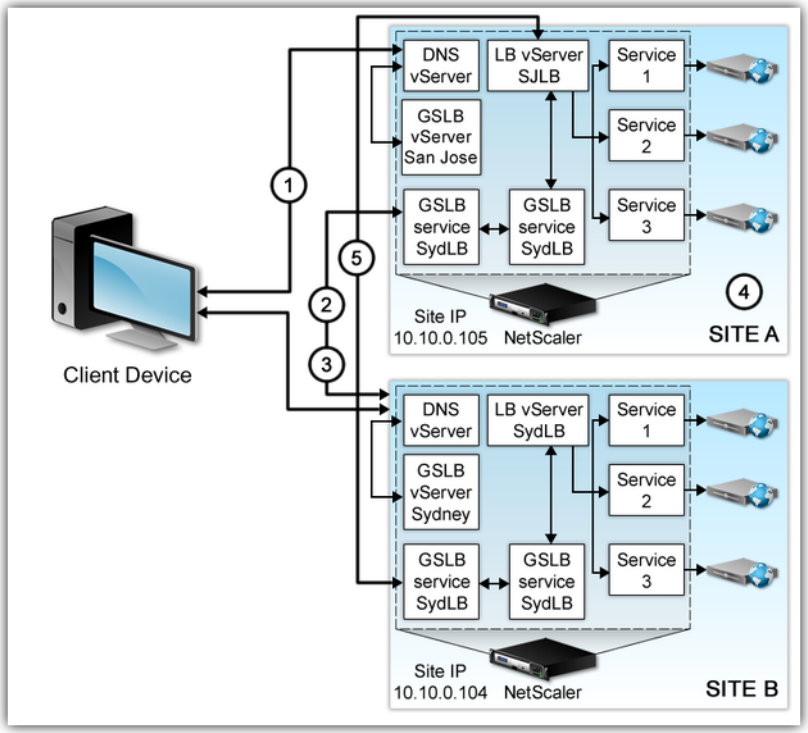
Unsecured RPC nodes use TCP port 3011. Secured RPC nodes use TCP port 3009.

 © Copyright 2012 Citrix Systems, Inc.

Source IP Address for an RPC Node

By default, the NetScaler system uses a SNIP address or MIP address as the source IP address for an RPC node, but you can configure the system to use a specific address. If neither a SNIP address nor a MIP address is available, the GSLB site cannot communicate with other sites. In such a scenario, you must configure either the NSIP address or a VIP address as the source IP address for an RPC node. A VIP address can be used as the source IP address of an RPC node only if the RPC node is a remote node. If you configure a VIP address as the source IP address and remove the VIP address, the system uses a SNIP address or a MIP address.

GSLB Site Communication Example



The figure and the following process illustrate GSLB site communication when a client uses www.example.com to make a DNS query:

1. A client or non-authoritative DNS server forwards a query to the ADNS/DNS virtual server on the NetScaler system. The www.example.com web site contains a DNS entry type GSLB domain pointing to the local GSLB virtual server.
2. The local DNS server/service forwards the query to the GSLB virtual server asking to respond with an IP address for the www.example.com site. The GSLB virtual server checks its metric table and returns the IP address of the load-balancing virtual server, which returns the best metric at the time of the client query.
3. The GSLB server forwards the IP address to the local DNS server.
4. The local DNS server responds to the client with the respective IP address.
5. The client makes a connection to the load-balancing virtual server using the IP address that has been returned.

 © Copyright 2012 Citrix Systems, Inc.

Customizing the GSLB Configuration

Once your basic GSLB configuration is operational, you can customize it by:

- Modifying the bandwidth of a GSLB service
- Configuring CNAME-based GSLB services
- Configuring static proximity
- Configuring dynamic RTT
- Configuring persistent connections
- Configuring dynamic weights for services
- Changing the GSLB method

You can also configure monitoring for GSLB services to determine their states.

Modifying Maximum Connections or Minimum Bandwidth for a GSLB Service

You can restrict the number of new clients that can simultaneously connect to a load-balancing or content-switching virtual server by configuring the maximum number of clients and/or the maximum bandwidth for the GSLB service that represents the virtual server.

To make these modifications in the configuration utility, go to **Configuration** > **GSLB** > **Services** , select the service you want to modify, and click **Open**. Then specify the maximum number of clients or maximum bandwidth to be consumed.

Using the command-line interface, enter:

```
set gslb service serviceName -maxClients positive_integer -
maxBandwidth positive_integer
```

Creating CNAME-based GSLB Services

To configure a GSLB service, you can use the IP address of the server or a canonical name of the server. If you want to run multiple services on different ports, you can use canonical names (CNAMES) for the services.

CNAME-based GSLB services are useful in a multilevel domain resolver configuration or in multilevel domain load balancing. Configuring a CNAME-based GSLB service can also help if the IP address of the physical server is likely to change. For example, you can have two entries in DNS as ftp.example.com and www.example.com for FTP services and HTTP services on the same domain, example.com.

If you configure CNAME-based GSLB services for a GSLB domain, when a query is sent for the GSLB domain, the NetScaler system provides a CNAME instead of an IP address. If the A record

for this CNAME record is not configured, the client must query the CNAME domain for the IP address. If the A record for this CNAME record is configured, the NetScaler system provides the CNAME with the corresponding A record (IP address). The NetScaler system handles the final resolution of the DNS query, as determined by the GSLB method.

In an IP-address-based GSLB service, the state of a service is determined by the state of the server that it represents. However, a CNAME-based GSLB service has its state set to UP by default; the VIP address or MEP are not used for determining its state. If a desktop-based monitor is bound to a CNAME-based GSLB service, the state of the service is determined according to the result of the monitor probes. You can bind a CNAME-based GSLB service only to a GSLB virtual server that has the DNS record type as CNAME.

There are certain limitations with CNAME-based GSLB services, namely that source IP hash and round robin are the only load-balancing methods supported. The static proximity method is not supported because a CNAME is not associated with an IP address, and static proximity can be maintained only according to the IP addresses. In addition, site persistence is not supported because the service referenced by a CNAME can be present at any third-party location.

 © Copyright 2012 Citrix Systems, Inc.

Changing the GSLB Method

Unlike traditional DNS servers that simply respond with the IP addresses of the configured servers, a NetScaler system configured for GSLB responds with the IP addresses of the services, as determined by the configured GSLB method. By default, the GSLB virtual server is set to the least-connection method. If all GSLB services show as DOWN, the NetScaler system responds with the IP addresses of all the configured GSLB services.

GSLB methods are algorithms that the GSLB virtual server uses to select the best-performing GSLB service. After the host name in the web address is resolved, the client sends traffic directly to the resolved service IP address. The NetScaler system provides the following GSLB methods:

- Round robin
- Least connections
- Least response time
- Least bandwidth
- Least packets
- Source IP hash
- Custom load
- Round trip time (RTT)
- Static proximity

For GSLB methods to work with a remote site, either MEP must be enabled or explicit monitors must be bound to the remote services. If MEP is disabled, RTT, least-connections, least-bandwidth, least-packets and least-response-time methods default to round robin. The static proximity and RTT load-balancing methods are specific to GSLB.

Proximity-based Load-Balancing Methods

Proximity-based load-balancing methods include:

- Dynamic network proximity
- Round trip time (RTT)
- Static proximity

Dynamic Network Proximity

Network proximity is a measure of how far an end user is from a data resource in terms of network distance or time. The GSLB feature monitors the real-time status of the network and directs the request of the client to the best site. GSLB uses a smoothed RTT metric to measure network proximity. Using the smoothed RTT metric could result in network congestion.

The RTT between the LDNS of the client and each of the participating sites is measured. The system then uses this metric to make its load-balancing decision. The DNS response generated by the system carries the IP address of the site closest to the LDNS of the client in terms of RTT.

Round Trip Time (RTT)

The GSLB solution uses its proprietary MEP to exchange metrics between participating sites. Using this protocol, the system learns the network metric information RTT of the LDNS of the client. When the LDNS of a client accesses the GSLB site for the first time, the RTT information is not available with the system. In such cases, the GSLB virtual IP address selects a site using the round-robin method and directs the client to this site. The system then starts calculating the RTT between the site and the LDNS. The system deployed at the participating site begins to calculate the RTT between the LDNS and the GSLB site. Periodically, the system participating in GSLB reports the RTT to other participating systems. When the DNS query is sent the next time, the system selects the best site using the network metrics.

The NetScaler system uses different mechanisms, such as ICMP echo request/reply (ping), UDP (DNS), and TCP, to probe the RTT metrics between the LDNS and the sites participating in the GSLB domain. First, a ping probe is performed to obtain the RTT. If the ping probe fails, the DNS probe is performed to calculate the RTT. If the DNS probe also fails, the TCP probe is performed.

Static Proximity

The static proximity feature uses an IP-address-based static location database. The database used to implement the static proximity method often contains all of the GSLB site locations. The NetScaler system uses this database to determine the proximity between the LDNS of the client and the GSLB sites. The NetScaler system responds with the IP address of the closest site that matches the proximity criteria.

The locations in the static GSLB database consist of an IP address range and up to six qualifiers for this range. The custom database is stored in `ns.conf` and a static third-party or NetScaler system database is stored in the `/var/netscaler/locdb` directory by default.

Implementing Proximity-based GSLB

When enabled, the proximity-based GSLB method allows the NetScaler system to make load-balancing decisions based on the proximity of the LDNS of the client to different sites. Proximity can be measured both statically and dynamically. The dynamic determination of proximity is based on the current network status, while the static determination of proximity is based on the geographic location of the LDNS of the client and the sites the client is accessing.

The main benefit of the proximity-based GSLB method is faster response time resulting from the selection of the closest available site.

 © Copyright 2012 Citrix Systems, Inc.

Static Proximity Configuration

For the static proximity method to work, you must either configure the NetScaler system to use an existing static proximity database populated through a location file or add custom entries to the static proximity database. After adding custom entries, you can set their location qualifiers. After configuring the database, you are ready to specify static proximity as the GSLB method.

A static proximity database is a UNIX-based ASCII file. Entries added to this database from a location file are called static entries. Only one location file can be loaded on a NetScaler system. Adding a new location file overrides the existing file. The number of entries in the static proximity database is limited by the configured memory in the NetScaler system.

The static proximity database can be created in the default format or in a format derived from commercially configured third-party databases. There is no strict enforcement of the database file format, except that the default file has format tags. The database files are ASCII files that use a comma as the field delimiter. There are differences in the structure of fields and the representation of IP addresses in the locations. The format parameter describes the structure of the file to the NetScaler system. Specifying an incorrect value for the format option can corrupt the internal data.

The default location of the database file is `/var/netscaler/locdb` and, in a high-availability setup, an identical copy of the file must be present in the same location on both NetScaler systems.

Dynamic RTT Configuration

Dynamic RTT is a measure of time or delay in the network between the client's LDNS server and a data resource. To measure dynamic RTT, the NetScaler system probes the client's LDNS server and gathers RTT metric information. The system then uses this metric to make its load-balancing decision. GSLB monitors the real-time status of the network and dynamically directs the client request to the datacenter with the lowest RTT value.

When a client's DNS request for a domain comes to the NetScaler system configured as the ADNS for that domain, the appliance uses the RTT value to select the IP address of the best-performing site to send it as a response to the DNS request.

The NetScaler system uses different mechanisms, such as ICMP echo request/reply (PING), TCP, and UDP to gather the RTT metrics for connections between the LDNS server and participating sites. The system first sends a ping probe to determine the RTT. If the ping probe fails, a DNS TCP probe is used. If that probe also fails, the system uses a DNS UDP probe.

The NetScaler system performs UDP probing on port 53 and TCP probing on port 80, and uses the proprietary MEP to exchange RTT values between participating sites. After calculating RTT metrics, the system sorts the RTT values to identify the datacenter with the best (smallest) RTT metric. If RTT information is not available, the NetScaler system selects a site by using the round-robin method and directs the client to the site.

To configure a GSLB virtual server for dynamic RTT, you specify the RTT load-balancing method. The NetScaler system regularly validates the timing information for a given local server. If a change

in latency exceeds the configured tolerance factor, the system updates its database with the new timing information and sends the new value to other GSLB sites by performing a MEP exchange. The default tolerance factor is 5 milliseconds (ms). The RTT tolerance factor must be the same throughout the GSLB domain. If you change it for a site, you must configure identical RTT tolerance factors on all NetScaler systems deployed in the GSLB domain.

Static and Dynamic RTT Configuration

You can configure the GSLB virtual server to use a combination of static proximity and dynamic RTT when you have some clients coming from an internal network like a branch office. You can configure GSLB such that the clients coming from the branch office or any other internal network are directed to a particular GSLB site that is geographically close to the client network. For all other requests, you can use dynamic RTT.

First, create two GSLB sites, local and remote. Then, for the local site, create a GSLB virtual server and GSLB services and bind the services to the virtual server. Then create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server at the local site. Finally, create a load-balancing virtual server with the same virtual server IP address as the GSLB service.

Once you have configured a basic GSLB setup, configure the GSLB virtual server to use static proximity for all traffic originating from an internal network and then use dynamic RTT for all other traffic.

 © Copyright 2012 Citrix Systems, Inc.

GSLB Persistence

GSLB persistence is a type of decision logic. When configuring GSLB persistence, you must be familiar with site persistence, cookie-based persistence, and connection proxy.

Site Persistence

Site persistence ensures that LDNS requests are sent to the same site and are not load balanced.

Set source IP address persistence by specifying the GSLB virtual server IP address and a unique identifier for the persistence type. In the configuration utility, go to **Configuration** > **GSLB** > **Services** .

Using the command-line interface, enter:

```
set gslb vserver gslbvip -
persistenceType IPAddress –persistenceID PositiveInteger
```

Cookie-based Persistence and Connection Proxy

Cookie-based persistence and connection proxy allow the HTTP-level persistence setting. Cookie-based persistence is configured on local GSLB services with the following arguments in the command-line interface:

- `-SitePersistence ConnectionProxy`
- `-cookieTimeout integer`
- `-CIP ENABLED cipheader`

Configuring of Persistence Connections

Persistence ensures that a series of client requests for a particular domain name is sent to the same datacenter instead of being load balanced. If persistence is configured for a particular domain, it takes precedence over the configured GSLB method. Persistence is useful for deployments that deal with e-commerce, where the server needs to maintain the state of the connection to track the transaction. To maintain the state of connection, you must configure persistence on a virtual server. With persistence configured, the NetScaler system selects a datacenter to process a client request and forwards the IP address of the selected datacenter for all subsequent DNS requests. If the configured persistence applies to a site that is DOWN, the NetScaler system uses a GSLB method to select a new site, and the new site becomes persistent for subsequent requests from the client.

The GSLB virtual server is responsible for DNS-based site persistence, and it controls the site persistence for a remote GSLB service. The NetScaler system supports persistence based on the source IP address or on HTTP cookies.

When you bring a physical service DOWN with a delay time, the physical service goes into the transition out of service (TROFS) state. Site persistence is supported as long as the service is in the TROFS state. That is, if the same client sends a request for the same service within the specified delay time after a service is marked TROFS, the same GSLB site (datacenter) services the request.

Configuring of Persistence-based on Source IP

When a DNS request is received at a datacenter in which source-IP-address-persistence is configured, the NetScaler system first looks for an entry in the persistence table and, if an entry for the LDNS server exists and the server mentioned in the entry is configured, the IP address of that server is sent as the DNS response. For the first request from a particular client, the NetScaler system selects the best GSLB site for the request and sends its IP address to the client. Since persistence is configured for the source IP address of the client, all subsequent requests by that client or another local DNS server in the same IP subnet are sent the IP address of the GSLB site that was selected for the first request.

For source-IP-address-based persistence, you must configure the same set of persistence identifiers on the GSLB virtual servers in all datacenters. A persistence identifier is a number used by the datacenters to identify a particular GSLB virtual server. A cookie transmits the persistence identifier, enabling the NetScaler system to identify the domain so that it can forward all appropriate requests to the same domain. When you enable persistence, the persistence information is also exchanged as part of metrics exchange. For the NetScaler system to support persistence across sites, you must enable persistence on the GSLB virtual servers of all participating sites. When you use source-IP-address persistence on the network identifier, you must configure a subnet mask. For any domain, persistence takes precedence over any other configured GSLB method.

Persistence-based on HTTP Cookies

The NetScaler system provides persistence at the HTTP-request level by using connection proxy and HTTP redirect. With these persistence methods, the system uses an HTTP cookie, also known as a site cookie, to reconnect the client to the same server. The NetScaler system inserts the site cookie in the first HTTP response.

The site cookie contains information about the selected GSLB service on which the client has a persistent connection. The cookie expiration is based on the cookie timeout configured on the NetScaler system. If the virtual server names are not identical on all the sites, you must use the persistence identifier. Cookies inserted are compliant with RFC 2109.

When the NetScaler system responds to a client DNS request by sending the IP address of the selected GSLB site, the client sends an HTTP request to that GSLB site. The physical server in that GSLB site adds a site cookie to the HTTP header and connection persistence is in effect. If the DNS entry in the client cache expires, and then the client sends another DNS query and is directed to a different GSLB site, the new GSLB site uses the site cookie present in the client request header to implement persistence.

If the GSLB configuration at the new site uses connection-proxy persistence, the new site creates a connection to the GSLB site that inserted the site cookie, proxies the client request to the original

 © Copyright 2012 Citrix Systems, Inc.

site, receives a response from the original GSLB site, relays that response back to the client, and closes the connection. If the GSLB configuration uses HTTP-redirect persistence, the new site redirects the request to the site that originally inserted the cookie.

Using Dynamic Weights for Services

In a network, there are usually servers that have a higher capacity for traffic than others. However, with a regular load-balancing configuration, the load is evenly distributed across all services even though different services represent servers with different capacities. To optimize your GSLB resources, you can configure dynamic weights on a GSLB virtual server. The dynamic weights can be based on either the total number of services bound to the virtual server or the sum of the weights of the individual services bound to the virtual server. Traffic distribution is then based on the weights configured for the services.

When you configure dynamic weights on the GSLB virtual server, requests are distributed according to the load-balancing method, the weight of the GSLB service, and the dynamic weight. The product of the weight of the GSLB service and the dynamic weight is known as the cumulative weight. Therefore, when you configure dynamic weight on the GSLB virtual server, requests are distributed on the basis of the load-balancing method and the cumulative weight. When you disable the dynamic weight for a virtual server, the numerical value is set to 1. This ensures that the cumulative weight is a non-zero integer at all times.

Dynamic weight can be based on the total number of active services bound to load-balancing virtual servers or on the weights assigned to the services. Consider a configuration with two GSLB sites configured for a domain and each site has two services that can serve the client. If a service at either site goes DOWN, the other server in that site has to handle twice as much traffic as a service at the other site. If dynamic weight is based on the number of active services, the site with both services active has twice the weight of the site with one service DOWN and therefore receives twice as much traffic.

Alternatively, consider a configuration in which the services at the first site represent servers that are twice as powerful as servers at the second site. If dynamic weight is based on the weights assigned to the services, twice as much traffic can be sent to the first site as to the second.

 © Copyright 2012 Citrix Systems, Inc.

Monitoring GSLB Services

When you bind a remote service to a GSLB virtual server, the GSLB sites exchange metric information including network metric information, which is the round-trip-time, and persistence information. If a metric exchange connection is momentarily lost between any of the participating sites, the remote site is marked as DOWN and load balancing is performed on the remaining sites that show as UP. When metric exchange for a site shows as DOWN, the remote services belonging to the site are marked as DOWN as well.

The NetScaler system periodically evaluates the state of the remote GSLB services by using either MEP or monitors that are explicitly bound to the remote services. Binding explicit monitors to local services is not required because the state of the local GSLB service is updated by default using the MEP. However, you can bind explicit monitors to a remote service. When monitors are explicitly bound, the state of the remote service is not controlled by the metric exchange.

By default, when you bind a monitor to a remote GSLB service, the NetScaler system uses the state of the service reported by the monitor. However, you can configure the NetScaler system to use monitors to evaluate services in the following situations:

- Always use monitors (default)
- Use monitors when MEP shows as DOWN
- Use monitors when remote services and MEP show as DOWN

The second and third of the above settings enable the NetScaler system to stop monitoring when MEP shows as UP. For example, in a hierarchical GSLB setup, a GSLB site provides the MEP information about its child sites to its parent site. Such an intermediate site may evaluate the state of the child site as DOWN because of network issues, though the actual state of the site shows as UP. In this case, you can bind monitors to the services of the parent site and disable MEP to determine the actual state of the remote service. This option enables you to control the manner in which the states of the remote services are determined.

Adding Monitors

To add a monitor, you specify the type and the port. You cannot remove a monitor that is bound to a service. You must first unbind the monitor from the service.

To add a monitor using the configuration utility, go to **Configuration** > **Load Balancing** > **Monitors** and click **Add**.

Using the command-line interface, enter:

```
add lb monitor name -type monitorType -destPort portNumber
```

Binding Monitors to a GSLB Service

Once you create monitors, you must bind them to GSLB services. When binding monitors to the services, you can specify a weight for the monitor. After binding one or more weighted monitors, you can configure a monitor threshold for the service. This threshold takes the service DOWN if the sum of the bound monitor weights falls below the threshold value.

In the configuration utility, you can set both the weight and the monitoring threshold at the same time that you bind the monitor. When using the command-line interface, you must issue a separate command to set the service's monitoring threshold.

To Bind a Monitor

Using the command-line interface, enter:

```
bind monitor name serviceName -state (Enabled | Disabled) -
weight positive_integer
```

Once complete, enter:

```
set gslb service ServiceName -monThreshold positive_integer
```

To bind a monitor using the configuration utility, use the following steps:

1. Go to **Configuration** > **GSLB** > **Services**.
2. Select the service to which you want to bind the monitor and click **Open**.
3. In the **Configure GSLB Service** dialog box, on the **Monitors** tab, select the monitor that you want to bind to the service, and click **Add**.
4. In the **Configured** table, you can select the newly assigned monitor and enter a new weight value.
5. To enable the monitor, verify that the **State** is selected.
6. Enter a threshold value in the **Monitor Threshold** text box and click **OK**.

 © Copyright 2012 Citrix Systems, Inc.

Monitoring GSLB Sites

The NetScaler system uses MEP or monitors to determine the state of the GSLB sites. You can configure a GSLB site to always use monitors, which is the default setting, use monitors when MEP shows as DOWN, or use monitors when both the remote service and MEP show as DOWN. In the latter two cases, the NetScaler system stops monitoring when MEP returns to the UP state.

Protecting the GSLB Setup Against Failure

You can protect your GSLB setup against failure of a GSLB site or a GSLB virtual server failure by configuring a backup GSLB virtual server, configuring the NetScaler system to respond with multiple IP addresses, or configuring a backup IP address for a GSLB domain. You can also divert excess traffic to a backup virtual server by using spillover.

Backup GSLB Virtual Server Configuration

Configuring a backup entity for a GSLB virtual server ensures that traffic to a site is not interrupted if the GSLB virtual server goes DOWN. The backup entity can be another GSLB virtual server, or it can be a backup IP address. With a backup entity configured, if the primary GSLB virtual server goes DOWN, the backup entity handles requests. To specify what should happen when the primary GSLB virtual server comes back UP again, you can configure a timeout period after which the primary takes over, or you can configure the backup entity to continue handling traffic until you manually enable the primary virtual server to take over by using the **disablePrimaryOnDown** option.

If you configure both the timeout and the **disablePrimaryOnDown** option for the backup entity, the backup session timeout takes precedence.

Configuration of a GSLB Setup to Respond to Multiple IP Addresses

A typical DNS response contains the IP address of the best-performing GSLB service. If you enable multiple IP response (MIR), the NetScaler system sends the best GSLB service as the first record in the response and adds the remaining active services as additional records. However, MIR is set to disabled by default. When disabled, the NetScaler system sends the best service as the only record in the response.

 © Copyright 2012 Citrix Systems, Inc.

Responding with an Empty Address Record When DOWN

A DNS response can contain either the IP address of the requested domain or an answer stating that the IP address for the domain is not known by the DNS server, in which case the query is forwarded to another name server. These are the only possible responses to a DNS query.

When a GSLB virtual server is disabled or in a DOWN state, the response to a DNS query for the GSLB domain bound to that virtual server contains the IP addresses of all the services bound to the virtual server. However, in this case you can configure the GSLB virtual server to send an empty DOWN response (EDR). When this option is set, a DNS response from a GSLB virtual server that is in a DOWN state does not contain IP address records, but the response code is successful. This prevents clients from attempting to connect to GSLB sites that are DOWN.

You must configure this setting for each virtual server to which you want it to apply.

Backup IP Address Configuration for a GSLB Domain

You can configure a backup site for your GSLB configuration. With this configuration in place, if all of the primary sites go DOWN, the IP address of the backup site is provided in the DNS response.

If a GSLB virtual server is active, that virtual server sends a DNS response with one of the active site IP addresses as selected by the configured GSLB method. However, if all the configured primary sites in the GSLB virtual server are inactive or in a DOWN state, the ADNS server or DNS server sends a DNS response with the IP address backup site.

 © Copyright 2012 Citrix Systems, Inc.

Diverting Excess Traffic to a Backup Virtual Server

Once the number of connections to a primary GSLB virtual server exceeds the configured threshold value, you can use the spillover option to divert new connections to a backup GSLB virtual server. This threshold value can be calculated dynamically or set manually. Once the number of connections to the primary virtual server drops below the threshold, the primary GSLB virtual server resumes serving client requests.

In addition, you can configure persistence with spillover. When you configure persistence, new clients are diverted to the backup virtual server if that client is not already connected to a primary virtual server. With persistence configured, connections that were diverted to the backup virtual server are not moved back to the primary virtual server after the number of connections to the primary virtual server drops below the threshold. Instead, the backup virtual server continues to process those connections until they are terminated by the end user. Meanwhile, the primary virtual server accepts new clients.

If the backup virtual server reaches the configured threshold and is unable to take any additional load, the primary virtual server diverts all requests to the designated redirect URL. If a redirect URL is not configured on the primary virtual server, subsequent requests are dropped. The spillover feature prevents the remote backup GSLB service from getting flooded with client requests when the primary GSLB virtual server fails. This occurs when a monitor is bound to a remote GSLB service, and the service experiences a failure that causes its state to go DOWN. The monitor continues to keep the state of the remote GSLB service UP, however, because of the spillover feature.

As part of the resolution to this problem, two states are maintained for a GSLB service, the primary state and the effective state. The primary state is the state of the primary virtual server and the effective state is the cumulative state of the virtual servers (primary and backup chain). The effective state is marked as UP if any of the virtual servers in the chain of virtual servers shows as UP. A flag that indicates that the primary VIP has reached the threshold is also provided. The threshold can be measured by either the number of connections or the bandwidth. A service is considered for GSLB only if its primary state is UP. Traffic is directed to the backup GSLB service only when all the primary virtual servers are DOWN. Typically, such deployments will have only one backup GSLB service. Adding primary and effective states to a GSLB service has the following effects:

- When source IP persistence is configured, the local DNS is directed to the previously selected site only if the primary virtual server on the selected site is UP and below threshold. Persistence can be ignored in the round-robin mode.
- If cookie-based persistence is configured, client requests are redirected only when the primary virtual server on the selected site is UP.
- If the primary virtual server has reached its saturation and the backup VIP(s) is absent or DOWN, the effective state is marked as DOWN.
- If external monitors are bound to an HTTP-HTTPS virtual server, the monitor decides the primary state.

© Copyright 2012 Citrix Systems, Inc.

- If there is no backup virtual server to the primary virtual server and the primary virtual server has reached its threshold, the effective state is marked as DOWN.

 © Copyright 2012 Citrix Systems, Inc.

Commonly-Used Deployment Scenarios for GSLB

GSLB is commonly used in the following deployment scenarios:

- GSLB for disaster recovery
- GSLB based on proximity
- GSLB based on scalability
- GSLB based on the number of Access Gateway end users
- GSLB for XenApp or XenDesktop

© Copyright 2012 Citrix Systems, Inc.

Implementing GSLB Failover for Disaster Recovery

Disaster recovery capability is critical because downtime is costly. A NetScaler system configured for GSLB forwards traffic to the least-loaded or best-performing datacenter. This configuration, referred to as an active-active setup, not only improves performance but also provides immediate disaster recovery by routing traffic to other datacenters if a datacenter that is part of the setup goes DOWN. Alternatively, you can configure an active-standby GSLB setup for disaster recovery only.

GSLB in an Active-Standby Datacenter Setup

A conventional disaster recovery setup includes an active datacenter and a standby datacenter. The standby datacenter is a remote site. When a failover occurs as a result of a disaster event that causes the primary active datacenter to be inactive, the standby datacenter becomes operational.

To configure disaster recovery in an active-standby datacenter setup, create an active datacenter by completing the following tasks:

- Add a local GSLB site
- Add a GSLB virtual server, which represents the active datacenter
- Bind the domain to the GSLB virtual server
- Add GSLB services and bind the services to the active GSLB virtual server

Then create the standby datacenter by completing the following tasks:

- Add a remote GSLB site
- Add a GSLB virtual server, which represents the standby datacenter
- Add GSLB services, which represent the standby datacenter, and bind the services to the standby GSLB virtual server
- Designate the standby datacenter by configuring the standby GSLB virtual server as the backup virtual server

Once you have configured the primary datacenter, replicate the configuration for the backup datacenter and designate it as the standby GSLB site by assigning a GSLB virtual server at that site as the backup virtual server.

GSLB in an Active-Active Datacenter Setup

An active-active GSLB deployment, in which both GSLB sites are active, removes any risk in having a standby datacenter. With such a setup, web or application content can be mirrored in geographically separate locations. This ensures that data is consistently available at each distributed datacenter.

 © Copyright 2012 Citrix Systems, Inc.

To configure GSLB for disaster recovery in an active-active datacenter setup, you must first configure the basic GSLB setup on the primary datacenter and then configure all other datacenters.

First create at least two GSLB sites. Then, for the local site create a virtual server and GSLB services and bind the services to the virtual servers. Next, create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server in the local site. Finally, at the local site create a load-balancing virtual server with the same virtual server IP address as the GSLB service.

Once you have configured the first datacenter, replicate the configuration for other datacenters that are part of the setup.

GSLB with Weighted Round Robin

When you configure GSLB to use the weighted round-robin method, weights are added to the GSLB services and the configured percentage of incoming traffic is sent to each GSLB site. For example, you can configure your GSLB setup to forward 80 percent of the traffic to one site and 20 percent of the traffic to another. After you do this, the NetScaler system will send four requests to the first site for each request that it sends to the second.

To set up the weighted round-robin method, first create two GSLB sites, local and remote. Next, for the local site create a GSLB virtual server and GSLB services and bind the services to the virtual servers. Configure the GSLB method as round robin. Next, create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server. Finally, create a load-balancing virtual server with the same virtual server IP address as the GSLB service.

Each service that represents a physical server in the network has weights associated with it. Therefore, the GSLB service is assigned a dynamic weight that is the sum of weights of all services bound to it. Traffic is then split between the GSLB services based on the ratio of the dynamic weight of the particular service to the total weight. You can also configure individual weights for each GSLB service instead of the dynamic weight.

If the services do not have weights associated with them, you can configure the GSLB virtual server to use the number of services bound to it to calculate the weight dynamically.

GSLB with Datacenter Persistence

Datacenter persistence is required for web applications that require maintaining a connection with the same server instead of having the requests load balanced. For example, in an e-commerce portal, maintaining a connection between the client and the same server is critical. For such applications, HTTP-redirect persistence can be configured in an active-active setup.

To configure GSLB for disaster recovery with datacenter persistence, you must first configure the basic GSLB setup and then configure HTTP-redirect persistence.

First create two GSLB sites, local and remote. Then, for the local site create a GSLB virtual server and GSLB services and bind the services to the virtual server. Next, create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server at the local site. Then, create a load-balancing virtual server with the same virtual server IP address as the GSLB service.

Finally, duplicate the previous steps for the remote configuration, or configure the NetScaler system to auto-synchronize your GSLB configuration.

 © Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

1. How do GSLB sites exchange metrics between one another?
    a. DNS
    b. SNMP
    c. RPC node
    d. ADNS
2. In which two circumstances would it be useful to create CNAME-based GSLB services? (Choose two.)
    a. The IP address of a physical server is likely to change.
    b. When using RTT as a load-balancing method.
    c. In a multilevel domain load balancing configuration.
    d. In a configuration with site persistence enabled.
3. Which two load-balancing methods are supported with CNAME-based GSLB services?
    a. Least Connections
    b. Source IP Hash
    c. Round Robin
    d. RTT

© Copyright 2012 Citrix Systems, Inc.

© Copyright 2012 Citrix Systems, Inc.

Module 9

Clustering

© Copyright 2012 Citrix Systems, Inc.

Overview

A NetScaler cluster is a group of NetScaler nCore systems working together as a single system image. Each system of the cluster is called a node. A NetScaler cluster can include as few as 2 or as many as 32 NetScaler nCore hardware or virtual systems as nodes.

The client traffic is distributed between the nodes to provide high availability, high throughput, and scalability.

After completing this module, you will be able to:

- Describe the NetScaler nCore Clustering feature.
- Describe the steps to set up a NetScaler nCore cluster.
- Manage and configure a NetScaler nCore cluster.
- Troubleshoot a NetScaler nCore cluster.

© Copyright 2012 Citrix Systems, Inc.

How Clustering Works

A NetScaler cluster is formed by grouping NetScaler systems that satisfy requirements specified in Hardware and Software Requirements. One of the cluster nodes is designated as a configuration coordinator (CCO). As the name suggests, this node coordinates all cluster configurations. The CCO also owns the cluster IP address which is the management address of the cluster. You configure the cluster by accessing the CCO through the cluster IP address.

You cannot configure an individual node by accessing it through the NetScaler IP (NSIP) address. Nodes accessed through the NSIP address are available in read-only mode. This means that you can only view the configurations and the statistics.

The configurations performed through the cluster IP address are propagated to the cluster nodes through a physical medium called the cluster backplane. The backplane is a logical grouping of physical connections, as are the client data plane and the server data plane.

The VIP addresses that you define on a cluster are available on all the nodes of the cluster (striped addresses). You can define MIP and SNIP addresses to be available on all nodes (striped addresses) or only on a single node (spotted addresses). The details of traffic distribution in a cluster depend on the algorithm used, but the same logical entities process the traffic in each case. Traffic is distributed only to nodes that are in the ACTIVE state, both administratively and operationally, and in the UP health state.

 © Copyright 2012 Citrix Systems, Inc.

Cluster Synchronization

The NetScaler configurations and the files that are available on the CCO are synchronized on all the other cluster nodes. This includes SSL certificates, licenses, and DNS information. To ensure that all nodes in a cluster are always in sync, NetScaler configurations on the CCO are synchronized to the other nodes of the cluster when:

- A node joins the cluster.
- A node rejoins the cluster.
- A sync command is executed on the CCO.

If there is a mismatch in the configurations available on the rejoined node and on the CCO, the node is synchronized by using one of the following techniques:

Full synchronization	If the difference between configurations exceeds 255 commands, all of the configurations implemented on the CCO are applied to the node that is rejoining the cluster. The node remains operationally unavailable for the duration of the synchronization.
Incremental synchronization	If the difference between configurations is less than or equal to 255 commands, only the configurations that are not available are applied to the node that is rejoining the cluster. The operational state of the node remains unaffected.

© Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

Match the following terms with the correct descriptions.

- client data plane
- cluster backplane
- server data plane

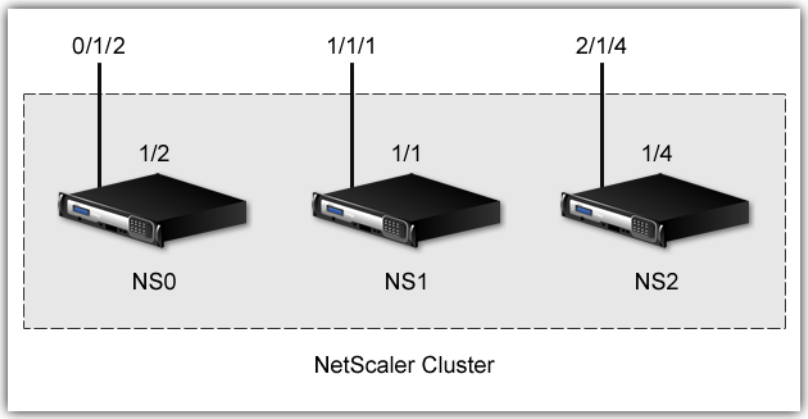
Description	Term
The logical grouping of the physical connections between the cluster nodes and the client-side connecting device.	
The logical grouping of the physical connections between the cluster nodes and the server-side connecting device.	
This allows the cluster nodes to communicate with one another and is the backbone of the cluster system.	

© Copyright 2012 Citrix Systems, Inc.

Cluster Connections

To identify the node to which an interface belongs, the standard NetScaler interface-naming convention is prefixed with a node ID. That is, the interface identifier c/u, where c is the controller number and u is the unit number, becomes n/c/u, where n is the node ID.

For example, in the following figure, interface 1/2 of Node 0 is represented as 0/1/2, interface 1/1 of Node 1 is represented as 1/1/1, and interface 1/4 of Node 2 is represented as 2/1/4.

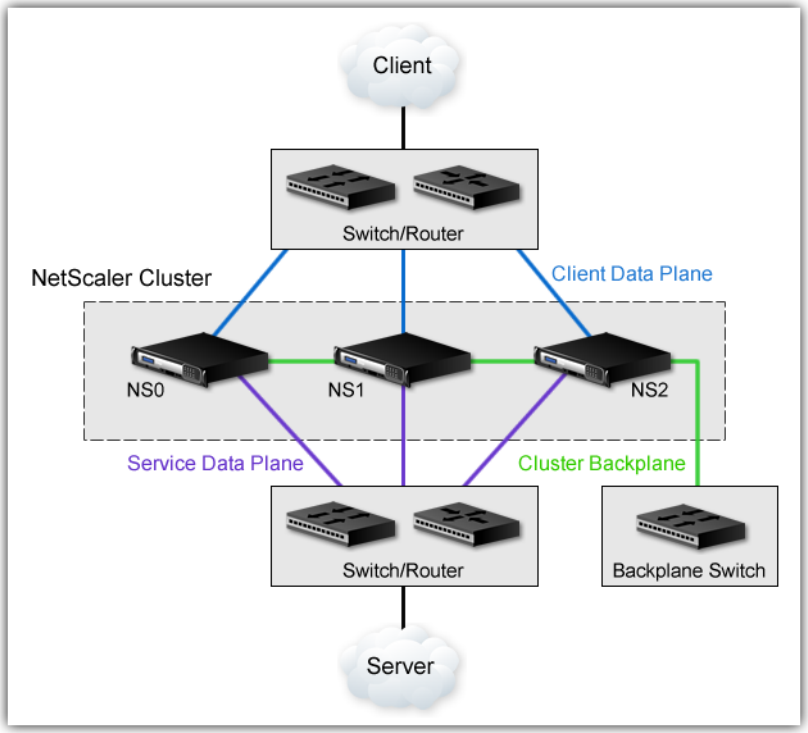


© Copyright 2012 Citrix Systems, Inc.

Cluster Communication Interfaces

The cluster communicates with the client through the physical connections between the cluster nodes and the client-side connecting device. The logical grouping of these physical connections is called the client data plane. Similarly, the cluster communicates with the server through the physical connections between the cluster nodes and the server-side connecting device. The logical grouping of these physical connections is called the server data plane.

In addition to communicating with the client and server through the client data plane and server data plane, respectively, the cluster nodes communicate with one another by using the cluster backplane. The backplane, which includes the physical connections between each cluster node and the backplane switch, is the backbone of the cluster system.



The figure shows the logical grouping of the physical connections to form the client data plane, server data plane, and cluster backplane.

 © Copyright 2012 Citrix Systems, Inc.

Striped and Spotted IP Addresses

In a clustered deployment, VIP, MIP, and SNIP addresses can be striped or spotted.

Striped IP address A striped IP address is active on all the nodes of the cluster. IP addresses configured on the cluster without specifying an owner node are active on all the cluster nodes.

Spotted IP address A spotted IP address is active on and owned exclusively by one node. IP addresses configured on the cluster by specifying an owner node are active only on the node that is specified as the owner.

The following table shows the NetScaler-owned IP addresses that can be striped or spotted:

NetScaler-owned IP address	Striped IP address	Spotted IP address
NSIP	No	Yes
Cluster IP address	No	No
VIP	Yes	No
MIP/SNIP	Yes	Yes (recommended)

The cluster IP address is neither a striped nor spotted IP address. It is a floating IP address that is owned by the CCO, which is not a fixed node.

Traffic Distribution

Client requests are sent to the NetScaler's striped VIP address and processed through these components:

1. Client data plane
2. Flow receiver
3. Flow processor

A traffic distribution mechanism configured on the client data plane selects one of the cluster nodes as the flow receiver.

The flow receiver receives the traffic, selects the node that must process the traffic, and directs the request to that node through the cluster backplane (unless the flow receiver selects itself as the flow processor).

The flow processor establishes a connection with the server. The server processes the request and sends the response to the mapped IP (MIP) address or subnet IP (SNIP) address that sent the request to the server.

 © Copyright 2012 Citrix Systems, Inc.

Cluster and Node States

Cluster-node classification includes three types of states: admin state, operational state, and health:

Admin state

An admin state is configured when you add the node to the cluster. It indicates the purpose of the node, which can be in one of the following states:

- ACTIVE. Nodes in this state serve traffic if they are operational and healthy.
- PASSIVE. Nodes in this state do not serve traffic but are in sync with the cluster. These nodes are useful during maintenance activity because they can be upgraded without removing the node from the cluster.
- SPARE. Nodes in this state do not serve traffic but are in sync with the cluster. Spare nodes act as backup nodes for the cluster. If one of the ACTIVE nodes becomes unavailable, the operational state of one of the spare nodes becomes ACTIVE and that node starts serving traffic.

Operational state

When a node is part of a cluster, its operational state can change to ACTIVE, INACTIVE, or UNKNOWN. There are a number of reasons for a node being in an INACTIVE or UNKNOWN state. Review the `ns.log` file or error counters to help determine the exact reason.

Health state

Depending on its health, a node can either be UP or NOT UP. To view the reasons for a node being in the NOT UP state, run the show cluster node command for that node from the cluster IP address.

Only nodes on which the admin state is ACTIVE, the operational state is ACTIVE, and the health status is UP can serve traffic. A cluster is functional only when a minimum of (n/2 +1) nodes, where n is the number of cluster nodes, are able to serve traffic.

© Copyright 2012 Citrix Systems, Inc.

NetScaler Cluster Setup

To set up a NetScaler cluster, you begin by setting up the cluster backplane. Then, you create the cluster by adding the first node, which becomes the initial configuration coordinator (CCO), and by assigning a cluster IP address to that node. Once the cluster IP address is defined on the CCO, you can add more nodes to the cluster.

Every system that you want to add to the cluster must:

- Be a NetScaler nCore appliance. Clustering of NetScaler Classic appliances is not supported.
- Be of the same platform type (physical appliance or VPX instance).
- Be of the same hardware type (for physical appliances).
- Be on the same subnet.
- Have the cluster license file.
- Have the same licenses (Standard, Enterprise, or Platinum, and any add-on licenses).
- Be of the same software version and build.
- Be initially configured and connected to a common client-side and server-side network.

Only systems that satisfy all the above criteria can be part of a NetScaler cluster.

To Set Up a Cluster Backplane

1. To set up the cluster backplane, perform the following steps for every node:
   a. Identify the network interface that you want to use for the backplane.
   b. Connect an Ethernet or optical cable from the selected network interface to the cluster backplane switch.

   For example, to use interface 1/2 as the backplane interface for node 4, connect a cable from the 1/2 interface of node 4 to the backplane switch.

To Add the Initial Configuration Coordinator

To create a cluster, you must create a cluster instance and configure a cluster IP address on the first system that you add to the cluster. This node is called the configuration coordinator (CCO). All cluster configurations are performed on this node by accessing it through the cluster IP address. The CCO is not fixed to one specific cluster node; it can change over time. For example, if the CCO goes DOWN, the cluster elects one of the other nodes as the new CCO, which then owns the cluster IP address.

Before you create the cluster, make sure that you have set up the backplane interface for that node.

1. Add a cluster instance. The cluster instance is an entity that identifies the cluster.

 © Copyright 2012 Citrix Systems, Inc.

```
add cluster instance <clId>
```

2. Add the NetScaler system to the cluster.

```
add cluster node <nodeId> <IPAddress> [-state <state>] [-
backplane <interface_name>]
```

3. Add the cluster IP address to this node.

```
add ns ip <IPAddress> <netmask> -type clip
```

4. Enable the cluster instance to create the cluster.

```
enable cluster instance <clId>
```

5. Save the configuration.
6. Warm restart the system.

To Add the Initial Configuration Coordinator (Configuration Utility)

To create a cluster by using the NetScaler configuration utility, perform the following steps:

1. Log on to a NetScaler system to add to the cluster.
2. In the navigation pane, expand **System** and then click **Cluster**.
3. In the details pane, under **Get Started**, click **Manage Cluster**.
4. In the **Cluster Configuration** dialog box, set the following parameters:
   - Cluster instance ID
   - Cluster IP address
   - Backplane
5. Click **Create**.
6. In the **Configure cluster instance** dialog box, ensure that the **Enable cluster instance** check box is selected.
7. In the **Cluster Nodes** pane, select the node, and click **Open**.
8. In the **Configure Cluster Node** dialog box, set the **State**.
9. Click **OK**, and then click **Save**.
10. Warm restart the system.

Cluster Node Addition

You can seamlessly scale the size of a cluster to include a maximum of 32 nodes. When a system is added to the cluster, the licenses on that system are checked against the licenses available on the CCO. If the licenses match, the system is added to the cluster. The existing configurations of the node are cleared, and the cluster configurations are synchronized with the node. This may cause an intermittent drop in traffic while the synchronization is in progress.

To add a node to a cluster, you must first configure the node on the cluster (adding the node) and then configure the cluster on the node (joining the cluster).

If you use the NetScaler command line, first log on to the cluster to add the node. Then, log on to that node and join the node to the cluster. If you use the configuration utility, log on to the cluster to add the node. The newly added node is automatically joined to the cluster. Alternatively, you can add the node from the command line and use the configuration utility to join the node to the cluster.

- Before you add the node, make sure that you have set up the backplane interface for that node.
- When you add a new node to a cluster that has only spotted IPs, the sync happens before spotted IP addresses are assigned to that node. In such cases, layer-3 VLAN bindings and static routes can be lost. To avoid this loss, either add a striped IP or add the layer-3 VLAN bindings and static routes on the NSIP of the newly added node.
- When a NetScaler system with pre-configured link aggregate (LA) channel is added to a cluster, the LA channels continue to exist in the cluster environment. The LA channel is renamed from LA/x to nodeId/LA/x, where LA/x is the LA channel identifier.

To Add a Node to the Cluster

To add a node to the cluster by using the NetScaler command-line interface, perform the following steps:

1. Log on to the cluster using the cluster IP address.
2. Add the NetScaler system to the cluster.
3. Save the configuration.
4. Log on to the newly added node.
5. Join the node to the cluster.
6. Save the configuration.
7. Warm restart the system.

 © Copyright 2012 Citrix Systems, Inc.

NetScaler Cluster Traffic Distribution Mechanisms

A NetScaler cluster supports these traffic distribution mechanisms:

Equal Cost Multiple Path (ECMP)	With the Equal Cost Multiple Path (ECMP) mechanism, the router has equal-cost routes to VIP addresses with the next hops as the active nodes of the cluster. The router uses a stateless hash-based mechanism to distribute traffic across the routes. To use ECMP, you must first enable the required routing protocol (OSPF, RIP, BGP, or ISIS) on the cluster IP address. You must add the interfaces to a VLAN, configure spotted IP addresses (unless you use the NSIP addresses of the cluster nodes) and enable dynamic routing on the spotted IP addresses, and configure the selected routing protocol on the ZebOS by using vtysh shell. You must perform equivalent configurations on the external connecting device. You must have detailed knowledge of routing protocols to use ECMP.
Linksets	Linksets must be used when some cluster nodes are not physically connected to the external network. In such a cluster topology, the unconnected cluster nodes use the interfaces specified in the linkset to communicate with the external network through the cluster backplane. Linksets are typically used in scenarios where the connecting devices of the external network have insufficient ports to connect the cluster nodes. For example, consider a three–node cluster where the upstream switch has only two ports available. Using linksets, you can connect two nodes to the switch and leave the third node unconnected.
Cluster Link Aggregation Group (CLAG)	A cluster link aggregation group (CLAG) is a group of interfaces of cluster nodes. It is an extension of NetScaler link aggregation. The only difference is that while link aggregation requires the interfaces to be from the same device, in CLAG the interfaces can be from different devices (cluster nodes). CLAG can be configured statically or dynamically. For example, consider a three-node cluster where all three nodes are connected to the upstream switch. A CLA channel (CLA/1) is formed by binding interfaces 0/1/2, 1/1/2, and 2/1/2.

© Copyright 2012 Citrix Systems, Inc.

To Configure Cluster Link Aggregation Group Statically

You can configure a CLA channel statically if you have control of the LA settings on the external connecting device.

1. Log on to the cluster.
2. Create a CLA channel.
3. Bind the required interfaces to the CLA channel. Ensure that the interfaces are not used for the cluster backplane.
4. Verify the CLA channel configurations.

 © Copyright 2012 Citrix Systems, Inc.

Dynamic Configuration for a Cluster Link Aggregation Group

To create a CLA channel dynamically, use Link Aggregation Control Protocol (LACP). Configuring a CLA channel dynamically is preferred when you do not have control of the LA configurations on the external connecting device.

When configuring CLAG, you should:

- Enable LACP (by specifying the LACP mode as either ACTIVE or PASSIVE).
- Choose an LACP key that will be used on each interface as part of the channel; the LACP value can be from 5 to 8.
- Specify the lag type as Cluster.

To configure a CLA channel dynamically by using the NetScaler command-line interface, type the following command for each interface that you want to add to the CLA channel:

```
set interface <onterface id> -lacpMode <lacpMode> -
lacpKey <lacpKey> -lagType Cluster
```

Test Your Knowledge

Indicate whether each statement is true or false.

Every system that you want to add to the cluster:

Statement	True or False
Must be a NetScaler Classic system.	
Must be the same platform type.	
Must be of the same hardware type (for physical appliances).	
Must be on the same subnet.	
Must have the cluster license file.	
Must have licenses but may be different types (Standard, Enterprise, or Platinum).	
May be of different software versions and builds.	
Must be initially configured and connected to a common client-side and server-side network.	

© Copyright 2012 Citrix Systems, Inc.

NetScaler Cluster Management

This section of the NetScaler clustering module includes the following cluster-management topics:

- Removing a cluster node
- Disabling a cluster node
- Viewing cluster instances and node details
- Synchronizing files on a cluster
- Upgrading or downgrading software on the cluster

Removing a Cluster Node

Removing a node from a cluster is a two-step process:

1. Remove the reference to the cluster instance from the node. This command internally executes the `clear ns config extended` command on that node. In addition, the MIP and SNIP addresses and all VLAN configurations (except default VLAN and NSVLAN) are cleared from the node.

> To remove a cluster instance, you must remove each node individually. When the last node is removed, the cluster IP addresses are deleted.

2. Remove the node from the cluster.

> If the node that you have removed is the current CCO, another node is made the CCO and the cluster IP address is assigned to that node.

To Disable a Cluster Node

You can temporarily remove a node from a cluster by disabling the cluster instance on that node. A disabled node is not synchronized with the cluster configurations and is unable to serve traffic.

1. To disable a cluster node by using the NetScaler command-line interface, perform the following steps:
   a. At the NetScaler command prompt of the node that you want to disable, type `disable cluster instance <clId>`, where <clId> identifies the cluster instance that you want to disable.
   b. To disable the cluster, type `disable cluster instance` on the cluster IP address.
2. To disable a cluster node by using the configuration utility, perform the following steps:
   a. Log on to the node that you want to disable.
   b. In the navigation pane, expand **System**, and then click **Cluster**.

c. In the details pane, under **Get Started**, click **Manage Cluster**.
d. In the **Configure cluster instance** dialog box, unselect the **Enable cluster instance** check box.
e. Click **OK**.
f. To disable the cluster instance on all the nodes, log on to the cluster and perform the above steps.

To View Cluster Instances and Node Details

You can view the details of the cluster instance and the cluster nodes from the cluster IP address.

1. To view details of a cluster instance by using the NetScaler command-line interface, perform the following steps:
   a. Type `sh cluster instance <clId>`, where <clId> is the integer that identifies the cluster instance whose details you want to view.

   Executing this command from the NSIP address of a non-CCO node displays the status of the cluster on this node.
2. To view details of a cluster node by using the NetScaler command-line interface, type `sh cluster node <nodeId>`, where <nodeId> is the integer that identifies the node whose details you want to view.
3. To view details of a cluster instance by using the configuration utility, perform the following steps:
   a. Log on to the cluster.
   b. In the navigation pane, expand **System**, and then click **Cluster**.
   c. In the details pane, under **Get Started**, click **Manage Cluster**.
   d. In the **Configure Cluster Instance** dialog box, view the details of the cluster.
4. To view details of a cluster node by using the configuration utility, perform the following steps:
   a. Log on to the cluster.
   b. In the navigation pane, expand **System**, click **Cluster**, and then click **Nodes**.
   c. In the **Cluster Nodes** list, view the node details. To get a more detailed view of the node, click the node.

To Synchronize Files on a Cluster

You can forcefully synchronize the configurations that are available on the CCO (full synchronization) to a specific cluster node.

1. To synchronize cluster configurations by using the NetScaler command-line interface, type: `force cluster sync`
2. To synchronize cluster configurations by using the configuration utility:

 © Copyright 2012 Citrix Systems, Inc.

a. Log on to the system on which you want to synchronize the CCO configurations.
b. In the navigation pane, expand **System** and then click **Cluster**.
c. In the details pane, below **Get Started**, click **Synchronize cluster configurations**.
d. Click **OK**.

Cluster Nodes Software Upgrades and Downgrades

All cluster nodes must be running the same software version. To upgrade or downgrade the software of a cluster, you must upgrade or downgrade the software on each node, one node at a time.

When the software on a node is upgraded or downgraded, the node is not removed from the cluster. The node continues to be a part of the cluster and serves client traffic uninterrupted, except for the downtime when the node restarts after it is upgraded or downgraded. However, due to software version mismatch among the cluster nodes, configuration propagation is disabled and is enabled only when all the cluster nodes are of the same version.

Since configuration propagation is disabled during upgrading or downgrading a cluster, you cannot perform any configurations through the cluster IP address during this time. However, you can perform node-level configurations through the NSIP address of individual nodes, as long as you perform the same configurations on all the nodes to keep them in sync.

© Copyright 2012 Citrix Systems, Inc.

NetScaler Cluster Troubleshooting

If a failure occurs in a NetScaler cluster, the first step in troubleshooting is to get information on the cluster instance and the cluster nodes by running the `show cluster instance <clId>` and `show cluster node <nodeId>` commands respectively.

If you are not able to find the issue by using the above two approaches, you can use one of the following:

- Isolate the source of the failure. Try bypassing the cluster to reach the server. If the attempt is successful, the problem is probably with the cluster setup.
- Check the commands recently executed. Run the history command to check the recent configurations performed on the cluster. You can also review the ns.conf file to verify the configurations that have been implemented.
- Check the `ns.log` files. Use the log files, available in the `/var/log/` directory of each node, to identify the commands executed, status of commands, and the state changes.
- Check the newnslog files. Use the newnslog files, available in the `/var/nslog/` directory of each node, to identify the events that have occurred on the cluster nodes. You can view multiple newnslog files as a single file by copying the files to a single directory and running the following command:

  ```
  nsconmsg -K newnslog-node<id> -K newnslog.node<id> -d current
  ```

If you still cannot resolve the issue, you can try tracing the packets on the cluster or use the `show tech support scope cluster` command to send the report to the technical support team.

To Trace the Packets of a Netscaler Cluster

The NetScaler operating system provides a utility called nstrace to get a dump of the packets that are received and sent out by an appliance. The utility stores the packets in trace files. You can use these files to debug problems in the flow of packets to the cluster nodes. The trace files must be viewed with the Wireshark application. For traces collected in native (.cap) mode, it is important to use the internal version of Wireshark, which can interpret native packets.

1. To trace packets of a NetScaler system, perform the following steps:
   a. Run the `start nstrace` command on the system. The command creates trace files in the `/var/nstrace/<date-timestamp>` directory. The trace file names are of the form `nstrace<id>.cap`.
   b. You can view the status by running the `show nstrace` command.
   c. Stop tracing the packets by running the `stop nstrace` command.
2. To trace packets of a cluster, run the `start nstrace` command on the cluster IP address. The command is propagated and executed on all the cluster nodes. The trace files are stored in individual cluster nodes in the `/var/nstrace/<date-timestamp>` directory. The trace file names are of the form `nstrace<id>_node<id>.cap`.

 © Copyright 2012 Citrix Systems, Inc.

3. To merge the trace files, at the NetScaler shell, type `nstracemerge.sh -srcdir <DIR> -dstdir <DIR> -filename <name> -filesize <num>` where:
   - `srcdir` is the directory from which the trace files are merged. All trace files within this directory are merged into a single file.
   - `dstdir` is the directory where the merged trace file is created.
   - `filename` is the name of the trace file that is created.
   - `filesize` is the size of the trace file.

Troubleshooting Joining and Syncing Issues in a Cluster

While joining a node to the cluster, I get the following message, "ERROR: Invalid interface name/number." What must I do to resolve this error?	This error occurs if you provided an invalid or incorrect backplane interface while using the `add cluster node` command to add the node. To resolve this error, verify the interface you provided while adding the node. Ensure that you have not specified the system's management interface as the backplane interface, and that the node ID bit of the interface is the same as the node ID. For example, if the node Id is 3, the backplane interface must be 3/<c>/<u>.
While joining a node to the cluster, I get the following message, "ERROR: Clustering cannot be enabled, because the local node is not a member of the cluster." What must I do to resolve this error?	This error occurs when you try to join a node without adding the node's NSIP to the cluster. To resolve this error, you must first add the node's NSIP address to the cluster by using the `add cluster node` command and then execute the join cluster command.
While joining a node to the cluster, I get the following message, "ERROR: Connection refused." What must I do to resolve this error?	This error can occur due to connectivity problems. The node cannot connect to the cluster IP address. Try pinging the cluster IP address from the node that you are trying to join. This error can also occur due to duplicate cluster IP address. Check to see if the cluster IP address exists on some non-cluster node. If it does, create a new cluster IP address and try rejoining the cluster.

While joining a node to the cluster, I get the following message, "ERROR: License mismatch between CCO and local node." What must I do to resolve this error?	The appliance that you are joining to the cluster must have the same licenses as the CCO. This error occurs when the licenses on the node you are joining do not match the licenses on the CCO. To resolve this error, run the `show ns hardware` and `show ns license` commands on both the nodes and compare the outputs.
What must I do when the configurations of a cluster node are not in sync with the cluster configurations?	In most cases, the configurations are automatically synchronized between all the cluster nodes. However, if you feel that the configurations are not synchronized on a specific node, you must force the synchronization by executing the `force cluster sync` command from the node that you want to synchronize.
The configurations across the cluster nodes are not synchronized. How can I ensure that the configurations are always in sync?	To ensure synchronization of the cluster configurations across the nodes, run the `save ns config` command after every configuration. Otherwise, the configurations might not be available on the cluster nodes when they restart.

Troubleshooting Common Issues

Why does the node state show INACTIVE when the node health shows UP?	A healthy node can be in the INACTIVE state for a number of reasons. A scan of `ns.log` or error counters can help you determine the exact reason.
How can I resolve the health of a node when its health shows NOT UP?	Node health NOT UP indicates that there are some issues with the node. To know the root cause, you must run the `sh cluster node` command. This command displays the node properties and the reason for the node failure.
When I run the `set vserver` command, I get the following message, "No such resource." What must I do to resolve this issue?	The `set vserver` command is not supported in clustering. The `unset vserver`, `enable vserver`, `disable vserver`, and `rm vserver` commands are also not supported. However, the `show vserver` command is supported.

 © Copyright 2012 Citrix Systems, Inc.

I cannot configure the cluster over a Telnet session. What must I do?	Over a telnet session, the cluster IP address can be accessed only in read-only mode. Hence, you cannot configure a cluster over a telnet session.
When configuring a cluster node, I get the following message, "ERROR: Session is read-only; connect to the cluster IP address to modify the configuration."	All configurations on a cluster must be done through the cluster IP address, and the configurations are propagated to the other cluster nodes. All sessions established through the NetScaler IP (NSIP) address of individual nodes are read-only.

Test Your Knowledge

Match the following errors with the probable cause of the error.

- Connection refused.
- Invalid interface name/number.
- License mismatch between CCO and local node.
- Clustering cannot be enabled, because the local node is not a member of the cluster.

Error	Probable Cause
	This error occurs if you provided an invalid or incorrect backplane interface while using the `add cluster node` command to add the node.
	This error occurs when you try to join a node without adding the node's NSIP to the cluster.
	This error can occur due to connectivity problems. The node cannot connect to the cluster IP address.
	This error occurs when the licenses on the node you are joining do not match the licenses on the CCO.

© Copyright 2012 Citrix Systems, Inc.

Module 10

Security and Authentication

© Copyright 2012 Citrix Systems, Inc.

Overview

Securing access to the NetScaler system requires a base comprehension of NetScaler internal and external communications. Regulating those communications along with system access is a key task for a NetScaler administrator.

Cloud computing has caused an increase in user interaction with the datacenter and in the need for web security. Citrix NetScaler protects data in the application layer and in transit and provides industry-leading security against unknown and known attacks.

Secure Sockets Layer (SSL) is a session-layer encryption and authentication protocol used with HTTP and other protocols to secure communications between clients and servers. SSL, which is widely used to support e-commerce web sites, specifies a method for a client and a server to exchange cryptographic information and to build a secure channel through which to send requests and responses.

To establish a secure SSL connection, however, an encryption key must be assigned to your application by a Certification Authority (CA) in the form of a certificate. Once it has a unique key of its own, you can establish a secure connection using the SSL protocol.

In addition, the processing of SSL transactions by a web server is CPU-intensive enough to negatively affect web application performance. Configuring SSL offload on the NetScaler can free the web servers from handling the SSL encryption and decryption tasks and maintain security without degrading web server performance. SSL offloading can also be used to optimize traffic, while maintaining the security of the private data.

After completing this module, you will be able to:

- Identify requirements to secure communications with SSL certificates.
- Create and upload an SSL certificate.
- Bind an SSL certificate key.
- Identify common virtual SSL server deployments.
- Configure advanced SSL options.
- Create appropriate servers and virtual servers.
- Identify items and entities that need to be secured in a NetScaler environment.
- Implement secure authentication for users to the NetScaler system and resources.
- Design and maintain access control lists to secure NetScaler communications.
- Configure NetScaler system options to protect against unauthorized attacks.

© Copyright 2012 Citrix Systems, Inc.

SSL

SSL is a protocol used to secure HTTP, TCP, and other types of traffic; it is the industry standard security technology for establishing encrypted links between a web server and a browser.

Transport Layer Security (TLS) abstracts the security protocol from any specific underlying service.

SSL/TLS encrypts the data using a certificate that has unique credentials identifying the owner and authenticating the identity of the certificate owner.

SSL Administration

An SSL certificate and key can be obtained for use on the NetScaler system using one of the following methods:

- Request certificate and key from a certificate authority (CA).
- Use an existing SSL certificate and key.
- Generate a new SSL certificate and key using the self-signing tools on the NetScaler system.

The process to create an SSL certificate is the same for the certificate authority or the NetScaler certificate tools:

1. Create a private key file.
2. Create a certificate signing request (CSR) using the private key file you created.
3. Submit the CSR to the certificate authority to create the certificate. If you are working with an external certificate authority, such as Verisign, submit the request to the CA which issued the certificate file. If you are using the NetScaler self-signing tools, the NetScaler system will use the CSR to generate the certificate.

Steps 1-3 can be skipped if a certificate and key file exist.

Use the NetScaler tools to create the key and request files when submitting a request to an external CA.

4. Install the certificate and key file on the NetScaler system.

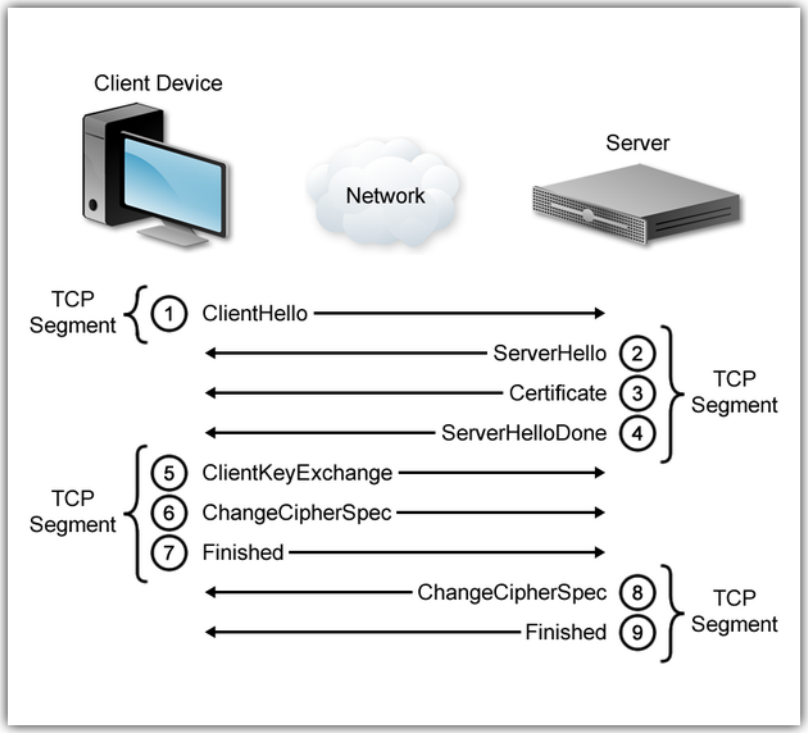
SSL and Policies

Like other features of the NetScaler system, SSL offload can be controlled through policies.

 © Copyright 2012 Citrix Systems, Inc.

SSL Session Process

For a client to establish a secure connection between a web browser and a server in most cases, a root certificate must be installed in the browser certificate store and on the server—which could be a web server, a service, or a NetScaler system. The following page describes the process in which a client and server initiate an SSL session.



The figure and the following process provide an overview of the process in which a client and server initiate an SSL session.

1. The client sends a Client-Hello with a set of supported cipher suites.
2. The server responds with a Server-Hello with the selected cipher suite.
3. The server sends its public key in the server certificate signed by the CA.
4. The server sends ServerHelloDone.
5. The client uses its root certificate to verify the signature of the server certificate. It then confirms that the certificate is still valid and that the subject name on the certificate matches the server name. The client generates a common key called the Premaster Secret.
6. The client transmits either the RSA-encrypted Premaster Secret or the Diffie Hellman (DH) parameters. Both options allow the client and the server to agree on the same Premaster Secret.

© Copyright 2012 Citrix Systems, Inc.

7. The client and the server, independently and in parallel, generate the same Master Secret using the Premaster Secret.
8. The client and the server generate session keys from the Master Secret.
9. The client and the server encrypt their communication using the session keys for the remainder of the SSL session.

SSL Keys

SSL Keys are strings of bits used to encrypt and decrypt messages. The NetScaler system supports RSA Algorithm keys and Digital Signature Algorithm (DSA) keys.

Keys are generated in the following situations:

- Before generating and submitting a certificate signing request to a certificate authority
- Before generating a self-signed certificate for testing purposes

The create key commands do not create objects in the NetScaler configuration; the commands create objects that are saved to the file system. Once generated, keys are saved by default to the [/nsconfig/ssl] directory in the NetScaler configuration.

When generating a key, you must designate the key size, or cipher key length, in bits. The current minimum standard for keys is 2048 bits, although the NetScaler system supports a key size as large as 4096 bits for RSA and DSA certificates. Higher key sizes result in more secure encryption algorithms.

Generate a key on the NetScaler system by specifying the key type, format, and encryption in the SSL node of the configuration utility.

Command-line interface syntax

RSA Key

```
create ssl rsakey keyFile bits [-exponents (3 | F4)]
[-keyform (DER | PEM)] [-des] [-des3] [-password string]
```

DSA Key

```
create ssl dsakey keyFile bits [-keyform (DER | PEM)]
[-des] [-des3] [-password string]
```

 © Copyright 2012 Citrix Systems, Inc.

Public Key Encryption

SSL and TLS make use of public and private key cryptography to verify the identities of the server and client and to set up the initial secure channel for the communication. Public-private key encryption is a form of asymmetric encryption, meaning that two separate but related keys are required.

Messages encrypted with the public key can be decrypted only using the private key. In turn, messages encrypted with the private key can be decrypted using only the public key. Public-private key encryption is a resource-intensive process.

© Copyright 2012 Citrix Systems, Inc.

Digital Certificates

Digital certificates are small files that contain public keys and verify the identity of the holder. Certificates are issued by a Certificate Authority (CA). A CA is an entity that issues certificates after verifying the identity of a server.

Generating certificate requests and applying certificates can sometimes be a complex process depending on the CA that you use. Citrix recommends to always use a common CA such as Thawte, VeriSign, or Network Solutions. These CAs are usually trusted by all Windows and Macintosh operating systems and therefore require less administrative overhead.

Although self-signed certificates save the cost of purchasing a certificate from a commercial CA, they have limitations. When using self-signed certificates, be aware that only internal clients will trust those certificates. Any external client that is not a member of the domain of the issuing server will have to add the trusted root before a trusted connection can be established.

Certificate Signing Request

Obtaining an SSL certificate from an authorized certificate authority requires the creation and submission of a certificate signing request (CSR).

Generate a CSR on the NetScaler system by specifying the key filename, format, and encryption. In the configuration utility, expand the SSL node. In the command-line interface, type the following command:

```
create ssl certReq [-keyFile |-fipsKeyName ]
[-keyform (DER | PEM) {-PEMPassPhrase}] -countryName -stateName -
organizationName
```

The following parameters are mandatory:

```
[-keyFile] -countryName -stateName -organizationName
```

The following additional parameters are optional:

```
[organizationUnitName][-localityName] [-commonName] [-
emailAddress] {-challengePassword} [-companyName]
```

For more information about generating a CSR, see Citrix article CTX109260 at *http://support.citrix.com*.

 © Copyright 2012 Citrix Systems, Inc.

SSL Certificates

The NetScaler system supports SSL certificates that are either self-signed or signed by a trusted certificate authority (CA). The NetScaler system can act as a certifying authority and provides options for generating a self-signed certificate for testing.

The NetScaler certificate tools can be used to generate the following certificate types:

- Root CA certificates
- Intermediate certificates
- Server certificates
- Client certificates

Root CA Certificates

Client devices use certificates to recognize which network entities to trust. Certificates are assigned to clients by a CA and carry an equivalent trust to that of the CA itself. A network entity requires a root_CA certificate to act as a CA and issue its own certificates to clients. This separate certificate is used to issue the client certificates. The root-CA certificate is critical, as it acts as the authority stamp on all client certificates that the CA will issue.

A NetScaler system can act as a CA and generate self-signed certificates by creating the root-CA certificate itself. In this scenario, clients must be configured to accept the NetScaler-created root-CA certificate to trust the client certificates that the NetScaler system will issue. To create a root-CA certificate, you must supply a key file and a certificate request.

Intermediate CA Certificates

Root CAs can assign certificates for an intermediate CA. An intermediate CA can then issue server certificates, client certificates, or certificates for other intermediate CAs. The intermediate CA is located between your root CA and enterprise CA and requires another level of validation, creating a verification chain.

Server Certificates

The NetScaler system must have a server certificate for an initial client SSL session. The server certificate verifies the server identity to the client. The certificate authority verifies that the company using the server certificate was authorized to issue a certificate for the domain. To create a server certificate, you must specify the certificate request, the root-CA, or intermediate CA certificate that should be used to generate the server certificate, details regarding the key format, and an output file for the CA serial number. For more information about creating a server certificate, see Citrix article CTX109260 at *http://support.citrix.com*.

© Copyright 2012 Citrix Systems, Inc.

Client Certificates

Client certificates are issued to an end user by a certification authority. The certificate can be used only by the client for whom the certificate was issued. The difference between server and client certificates is that server certificates provide encryption and security functionality, whereas client certificates provide end user authentication functionality.

Certificate Generation

Generate a Certificate

Generate a certificate on the NetScaler system by specifying the certificate format, certificate type, CA certificate file format, CA key file name, CA key file format, CA key encryption, and CA serial number file.

The CA parameters are optional.

In the configuration utility, generate a certificate under the SSL node.

From the command line, use the following command:

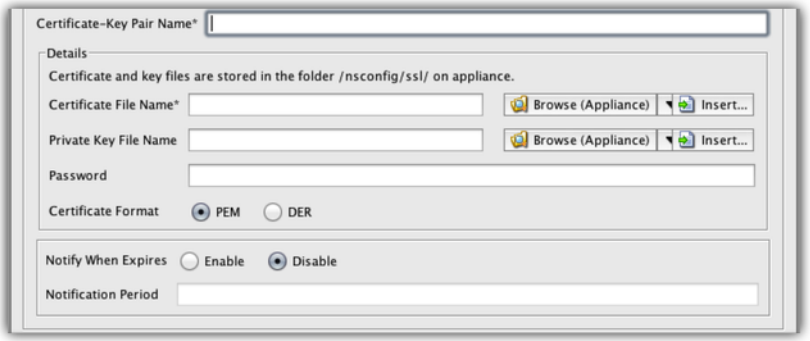
```
create ssl cert certFile reqFile certType [-
keyFile input_fileName]
[-keyForm (DER | PEM)] [-days positive_integer] [-
certForm (DER | PEM)] [-CAcert input_fileName]
[-CAcertForm (DER | PEM)] [-CAkey input_fileName] [-
CAkeyForm (DER | PEM)]
[-CAserial output_fileName]
```

For more information about replacing the default certificate of the NetScaler with a trusted CA certificate that matches the hostname of the NetScaler, see Citrix article CTX122521 at *http://support.citrix.com*.

Certificate Key Pairs

For SSL processing to occur, an certificate key entity must be bound to the virtual server. A certificate key entity is an integral element of the SSL encryption and decryption process, which is used during the SSL handshake to determine the cipher that will be used for SSL processing and also to establish the identity of the SSL server.

 © Copyright 2012 Citrix Systems, Inc.



Certificate Key Pair

Before a client certificate can be used for SSL processing, it must be paired with its corresponding certificate key entity which resides on the NetScaler system. This certkey pair is then bound to the client connection for SSL processing.

Adding a Certificate Key Pair

Add a certificate key entity on the NetScaler system by specifying the certificate file name, the private file name, the password, and the certificate format:

Configuration Utility location	Under the SSL node
Command-line syntax	Use the following command:

```
add ssl certkey certkeyName -cert fileName[(-key fileName [-
password]) | -fipskey string]
[-inform (DER | PEM)][-expiryMonitor (ENABLED | DISABLED)][-
notificationPeriod positive_integer]
```

For more information about adding a certificate key pair, see Citrix article CTX109260 at *http://support.citrix.com*.

Bind a Certificate Key Pair to a Virtual Server

Bind a certificate key pair by specifying the virtual server name and the key pair name on the NetScaler system

Configuration Utility location	under the SSL node
Command-line syntax	Use the following command:

```
bind ssl vserver -certkeyname
```

Before the certificate key pair can be used, you will need to physically place the file on the NetScaler and then add it to the system for binding to the appropriate entities. If you are using the command-line interface, you will need to transfer the certificate to the NetScaler, typically by the way of SCP protocol, to the directory.

Certificate Revocation List (CRL)

A Certification Revocation List (CRL) is a list of the serial numbers of certificates that have been revoked and are no longer valid. Users should never accept this type of certificate to establish access to a secured resource.

If client or server SSL certification validation is required, the NetScaler system can perform validation against a local certificate revocation list in a local file.

This list can be updated on a specified interval automatically. Supported update methods include HTTP and LDAP. Validation is not the default setting and must be enabled.

Certificate Updates

You may need to update or replace a certificate on the NetScaler system for a variety of reasons:

- Certificate is expired or expiring soon.
- Default certificate needs to be replaced with trusted CA certificate.
- File name or private key have changed or have been compromised.

A certificate can be updated from the configuration utility or the command line. In the configuration utility navigation pane, expand the **SSL** node, click **Certificates**, select the certificate you want to update, click **Update**, and update the certificate where necessary. For notification about when a certificate expires, enable the **Notify when expires** checkbox.

SNMP monitoring can also be used to send alerts when a certificate is about to expire. Citrix recommends to run reports on all certificate use and expiration dates to ensure availability of resources.

For more information about updating or replacing an SSL certificate, see Citrix article CTX109711 at *http://support.citrix.com*.

 © Copyright 2012 Citrix Systems, Inc.

Certificates can also be updated from Command Center. See Citrix eDocs for Command Center for more information.

For a demonstration about how to update certificates on the NetScaler system, see Citrix TV video 2960 at *http://citrix.com/tv*.

NetScaler System Communication

To secure NetScaler system communication, you must first trace the external and the internal communication that occurs when the NetScaler system is deployed.

NetScaler communication types and layers must be secured. The NetScaler system uses layer 2 for internal, external, and management communication. Access control lists also communicate on layer 2, layer 3, and layer 4.

Typical communication types include:

System-to-system communication	High availability is a type of system-to-system communication. High-availability communication occurs by default on TCP port 3010 and 3011. High-availability communication is secure on TCP port 3008 and 3009 and is used for command synchronization and command propagation. UDP port 3003 is used for the high-availability heartbeat and is a health-check industry standard. GSLB is a type of system-to-system communication that occurs by default on TCP port 3011. GSLB communication is secure on TCP port 3009 and is used for GSLB metric exchange protocol communication.
Management communication	The command-line interface is a type of management communication involving SSH on TCP port 22 (Secure Telnet). The command-line interface is accessed through Telnet on TCP port 23, which is not secure and is disabled by default. The configuration utility and GUI are a type of management communication that includes the front end and the Java applet. The front end is on TCP port 80, which is the default port. It is secure on TCP port 443. The Java applet is on TCP port 3010, which is the default port. It is secure on TCP port 3008.
SNMP request and response traffic communication	SNMP request and response traffic (polling) is a type of communication that occurs on UDP port 161. SOAP XML API occurs on TCP port 80 using HTTP, and it is secure on TCP port 443 using HTTPS.

Citrix recommends limiting communication to the NSIP address to management access.

 © Copyright 2012 Citrix Systems, Inc.

Port	Type	Use	Secure
22. TCP	HTTP	SSH	Secure
80. TCP	HTTP	GUI web front end, XML-API	Non-secure
161. UDP	HTTP	SNMP request/response traffic (polling)	Non-secure
162. UDP	HTTP	SNMP traps	Non-secure
443. TCP	HTTPS	GUI web front end, XML-API	Secure
3003. UDP	HTTP	HA Heartbeat	Non-secure
3008. TCP	HTTPS	GUI Java applet, HA Communication	Secure
3009. TCP	HTTPS	HA Communication, GSLB MEP communication	Secure
3010. TCP	HTTP	GUI Java applet, HA Communication	Non-secure
3011. TCP	HTTP	HA Communication, GSLB MEP communication	Non-secure

© Copyright 2012 Citrix Systems, Inc.

Deployment Scenarios

The SSL requirements for a particular environment depend on how SSL will be deployed. The following scenarios are the most common:

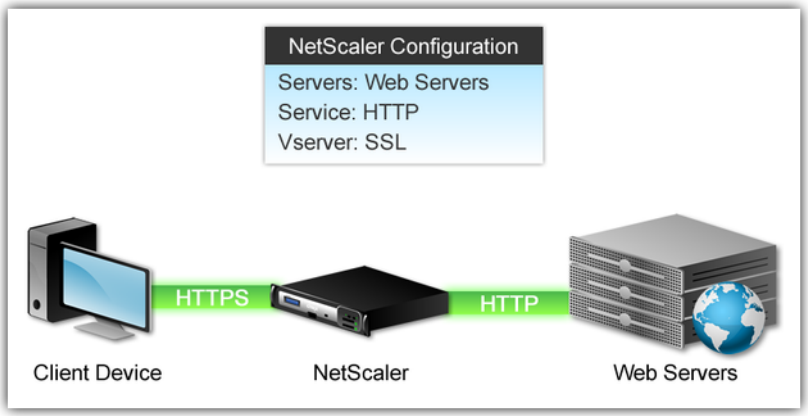
Front-end SSL with Back-end HTTP	Front-end SSL with back-end HTTP refers to a network configuration that allows the NetScaler system to offload SSL encryption and decryption and pass traffic to web servers in plain text.
Front-end SSL with Back-end SSL	Front-end SSL with back-end SSL refers to a network configuration that allows the NetScaler system to pass traffic to web servers in encrypted mode. Although the NetScaler system does not fully offload SSL processing, it can provide a benefit through SSL connection multiplexing or the use of a smaller cipher.
Front-end TCP over SSL with Back-end TCP	Front-end TCP over SSL with back-end TCP refers to a network configuration that allows the NetScaler system to offload SSL encryption and decryption for client connections, as well as pass any TCP traffic to servers in plain text. An example of this type of deployment scenario is providing secure access for LDAP directories.

Front-end SSL with Back-end HTTP Requirements

The figure and the following list provide an overview of the requirements necessary for a front-end SSL with back-end HTTP configuration:

- An installed certificate
- A load-balancing virtual server using the SSL protocol
- One or more HTTP services associated with back-end web servers

 © Copyright 2012 Citrix Systems, Inc.



Securing Traffic Between a Client and the NetScaler System

Secure the traffic between a client and the NetScaler system using the following process:

1. Export an existing key and certificate or generate a new key, CSR, and Certificate.
2. Convert the certificate file to PEM or DER using OpenSSL, if applicable.
3. Upload the certificate and key to the NetScaler system using an SCP or SFTP application.

> You can also upload the certificate using the configuration utility.

4. Create a certkey entity to install the certificate and key pair on the NetScaler system.
5. Create servers that point to the back-end web servers.

> If the servers have been created previously, this step is not necessary.

6. Create an HTTP service associated with the web servers.

> If the service has been created previously, this step is not necessary.

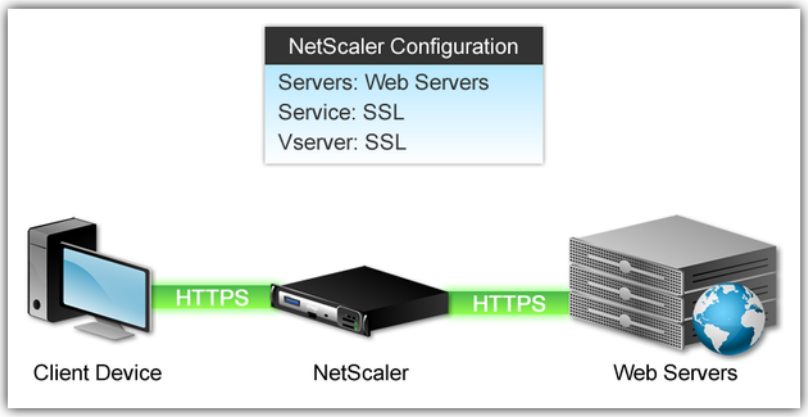
7. Create a load-balancing SSL virtual server.
8. Bind the virtual server to the services.
9. Bind the certkey to the virtual server.

Front-end SSL with Back-end SSL Requirements

The figure and the following list provide an overview of the requirements necessary for a front-end SSL with back-end SSL configuration:

- An installed certkey for traffic between the NetScaler system and client
- A load-balancing virtual server set to use the SSL protocol
- An SSL service or services associated with back-end web servers

For a more secure configuration, install a certkey on the SSL server for traffic between the NetScaler and the back-end web servers. Since the service type is HTTPS, the certificate is installed on the web servers.



Securing Traffic Between the NetScaler and the Server

Secure the traffic between the NetScaler system and web servers using the following process:

1. Install certificates to secure traffic between the client and the NetScaler system, as well as between the NetScaler system and the back-end web servers.
2. Create servers that point to the back-end web servers.

> If the servers have been created previously, this step is not necessary.

3. Create SSL services associated with the web servers.

> If the services have been created previously, this step is not necessary.

4. Create an SSL virtual server for the web servers.

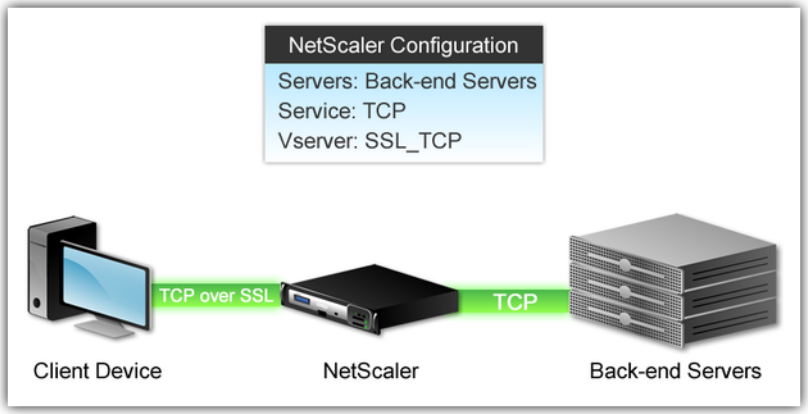
 © Copyright 2012 Citrix Systems, Inc.

5. Bind the appropriate services or server groups to the virtual server.

Front-end SSL_TCP with Back-end TCP Requirements

The figure and the following list provide an overview of the requirements necessary for a front-end SSL_TCP with a back-end TCP configuration:

- An installed certkey
- A load-balancing virtual server using the SSL_TCP protocol
- A TCP service or services associated with back-end web servers



Securing TCP Traffic Between a Client and the NetScaler System

Secure TCP traffic between a client and the NetScaler system, as well as pass TCP traffic to server using the following process:

1. Install certificates to secure traffic between the client and the NetScaler system.
2. Create servers that point to the back-end web servers.

> If the servers have been created previously, this step is not necessary.

3. Create TCP services associated with the servers.

> If the services have been created previously, this step is not necessary.

4. Create an SSL_TCP virtual server.
5. Bind the appropriate services to the virtual server.
6. Bind the appropriate certkey to the SSL_TCP virtual server.

SSL_Bridge

The SSL_BRIDGE functionality allows all secure traffic to be bridged transparently and directly to the back-end web server. The system does not terminate or offload this traffic. In this scenario, the web server must handle all SSL-related processing.

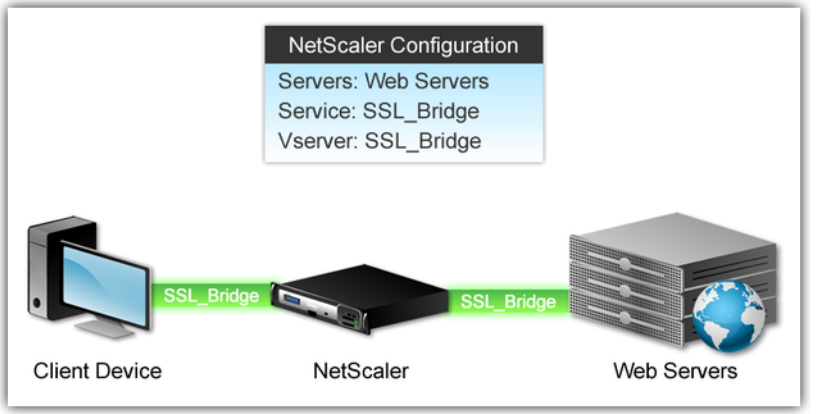
> This configuration should be used only if an acceleration unit (for example, a PCI-based SSL accelerator card) is installed in the web server to handle the SSL processing overhead.

In an SSL_BRIDGE setup, the system is configured to load balance and maintain server persistency for secure requests using the SSL Session-ID. Other features, such as content switching, Sure Connect, and cache redirection do not function because the traffic passing through the NetScaler system is encrypted. Because the system does not carry out any SSL processing in an SSL_BRIDGE setup, it is not possible to bind a certkey to the SSL_BRIDGE virtual server.

SSL_Bridge Requirements

The following requirements are necessary for an SSL_Bridge configuration:

- A load-balancing virtual server using the SSL_Bridge protocol
- A SSL_Bridge service or services associated with back-end web servers



 © Copyright 2012 Citrix Systems, Inc.

SSL Termination Points

The NetScaler system can also provide protection against encrypted attacks by terminating and offloading SSL. By terminating the SSL, the NetScaler system can inspect and protect all traffic going to the servers in addition to switching it intelligently. Without this benefit, attacks can be wrapped in SSL and delivered securely to the server.

To properly configure the NetScaler system, you must determine the SSL termination points. SSL transactions may be terminated on one of the following devices:

- Citrix NetScaler
- Citrix Application Firewall
- Network firewall
- Web server
- Access Gateway

The following termination methods can be used on the NetScaler system or on the servers:

- Terminating SSL on NetScaler System
- Terminating SSL on the servers

Terminating SSL on the NetScaler System

Terminating SSL on the NetScaler system has several benefits:

- The server is freed from the extensive CPU cycles required for the handshake, encryption, and decryption processes.
- Switching decisions can be made based on request content.
- Denial-of-service attack protections can be implemented.

Terminating SSL on the Servers

The primary benefit to terminating SSL on the servers is the ability to deploy a NetScaler system without any changes to the server configuration. Furthermore, if the NetScaler system is intercepting the connection, SYN attack prevention benefits can be maintained. However, the server will require significant processing power to handle the SSL transactions.

In most cases, SSL sessions will be terminated on the NetScaler system. In critical banking or financial transactions, SSL encryption must continue through to the servers to ensure that the transaction cannot be compromised under any condition.

© Copyright 2012 Citrix Systems, Inc.

Features and Benefits

The NetScaler SSL implementation supports a full feature set and is interoperable with all common SSL clients.

Feature	Benefit
Integrated or 1-arm	Scalability
SSL protocol support (SSLv2, SSLv3, TLSv1)	Rich protocol suite
Cipher suite support: • Key-exchange: RSA, DSS, DH [Max key: 4096 bits] • Encryption: RC4, DES, 3DES, RC2 • MAC: SHA, SHA-1, MD5	Rich cipher suite—The level of security can be adjusted based on client needs with a full range of key exchange, encryption, and authentication protocols.
Client authentication	• Authenticity of the end user requesting the web object • Web access on a user-by-user basis
Certificate Revocation List (CRL)/Online Certificate Status Protocol (OCSP)	Greater security, with ability to block clients with revoked certificates
• As many as 380,000 secure transactions every second • Up to 11 Gbps bulk encryption rate Features and performance depend on your specific NetScaler platform. For more information, see the NetScaler Datasheet at *http://www.citrix.com*	• Better performance • Quicker response time • Handling of traffic surges such as flash crowds
Centralized certificate/key management	Ease of management
Easy drop-in installation	• No changes are needed to the web servers. • No additional hardware or software is needed.

 © Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

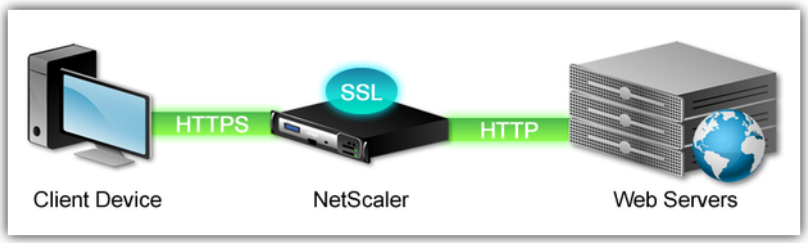
1. Which two options are valid means of obtaining a server certificate for use in a production environment? (Choose two.)
    a. Request a certificate from a certificate authority (CA).
    b. Use an existing certificate from a back-end web server.
    c. Install a standard server certificate issued with most web browsers.
    d. Generate a new SSL certificate and key using the self-signing tools on the NetScaler system.
2. Which benefits result from terminating SSL transactions on the NetScaler system rather than on the server? (Choose two.)
    a. The server is freed from extensive CPU cycles.
    b. SSL encryption can be passed to the back-end server.
    c. A NetScaler system can be deployed without changes to the server configuration.
    d. SSL transactions terminated on a NetScaler are not as easily compromised as those terminated on a server.
3. Which characteristic is shared by all access control lists?
    a. Disabled by default
    b. Enabled by default
    c. Compare outgoing packets of data
    d. Are not retained in the configuration

© Copyright 2012 Citrix Systems, Inc.

SSL Offload Overview

The SSL offload feature of the NetScaler system handles the CPU-intensive SSL encryption and decryption processing, allowing the web servers to dedicate more processing power to content requests. The SSL offload feature increases the performance of web sites that carry out SSL transactions.

The figure provides an overview of a strict SSL offload scenario in which all SSL encrypted communication between the web servers and the client is handled by the NetScaler system. Communication between the NetScaler system and the back-end server is unencrypted, providing load reduction on the server and allowing the server to focus on performing the application role instead of on managing SSL encryption/decryption processes.

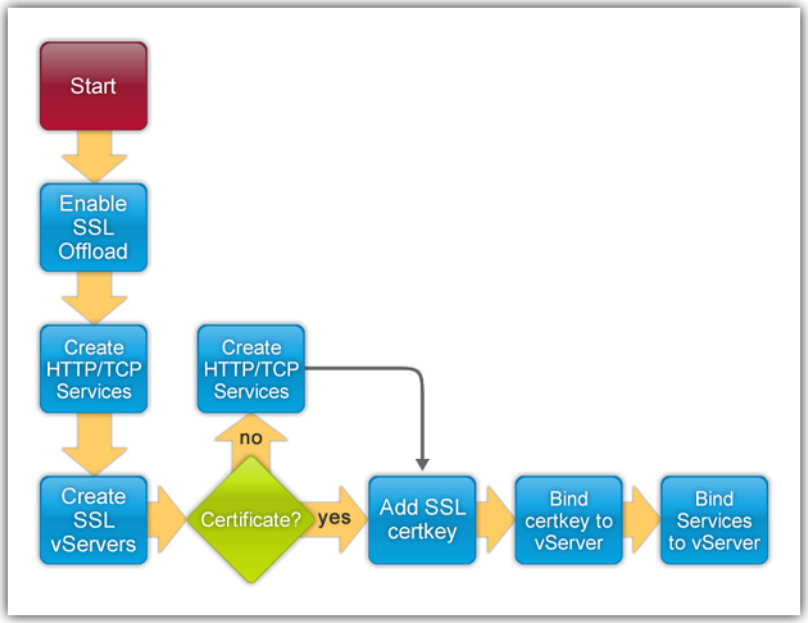


In other scenarios, the security of the data needs to be maintained end-to-end and should not be transmitted in a non-encrypted format. Encryption will occur between the client and the NetScaler system and also between the NetScaler system and the server. The server still needs to perform work to encrypt the data, but the NetScaler system is able to optimize the traffic and improve efficiencies by reusing the SSL connections.

Configuring SSL Offload

SSL must be enabled to configure the NetScaler system for SSL Offload.

 © Copyright 2012 Citrix Systems, Inc.



The figure and the following steps provide an overview of the process used to configure SSL Offload on the NetScaler system.

1. Enable SSL Offload on the NetScaler system.
2. Create HTTP/TCP/SSL services on the NetScaler system.
3. Create SSL virtual servers on the NetScaler system.
4. Use an existing key and certificate or obtain a key and certificate.
5. Upload the key and certificate to the NetScaler system.
6. Create the certkey entity on the NetScaler system from the uploaded key and certificate.
7. Bind the certkey to the SSL or SSL_TCP virtual server.
8. Bind the services to the virtual server.

Offload Performance

The NetScaler system supports extremely high-performance SSL encryption and session creation. For example, the NetScaler MPX platforms support:

- As much as 11 Gbps of bulk encryption
- As much as 380,000 SSL handshakes every second

The NetScaler system can offload a significant portion of the total application load from the servers it supports as well as switch the resulting SSL traffic. This offload cannot be performed efficiently by traffic management systems. If the traffic management system is not decrypting the data, it is

© Copyright 2012 Citrix Systems, Inc.

unable to make load-balancing decisions based on application data, such as HTTP header information or cookies. Instead, legacy traffic management systems must treat all of the traffic like raw IP traffic and make decisions based on source and destination IP addresses. By decrypting the traffic as it enters the NetScaler system, SSL processing can be offloaded from application servers, allowing the robust traffic management functionality of the NetScaler system to come into play. SSL Offload to a NetScaler also allows for simpler certificate management. Fewer certificates are needed, thus reducing cost.

 © Copyright 2012 Citrix Systems, Inc.

Advanced SSL Settings

Three commonly used advanced SSL configuration options include:

- SessionReuse
- ClientAuthentication
- SNI

Other SSL settings include:

- SSL redirect
- SSLv2 protocol
- SSLv2 redirect

SSL Redirect

Enabling SSL Redirect causes the NetScaler system to convert any HTTP 302 redirect responses from back-end servers to an HTTPS redirect. SSL Redirect is useful if HTTPS is being used between the client and the NetScaler system, but HTTP is being used between the NetScaler system and back-end server.

For example, without SSL Redirect enabled, when a client with an established SSL session follows a URL that begins with HTTP://, the SSL session will be broken. By enabling SSL Redirect, the NetScaler system will intercept the redirect and append an S to the URL protocol, so that the URL begins with HTTPS:// instead. The NetScaler system maintains the secure channel without requiring any changes to the back-end server.

For more information about SSL Redirect, see Citrix eDocs at *http://edocs.citrix.com*.

SSLv2 Protocol and Redirect

SSLv2

SSL protocol version 2 (SSLv2) is disabled by default because of known vulnerabilities. However, if you know that clients connecting to the network support only SSLv2, it can be enabled through the Configure SSL Params menu of the configuration utility.

SSLv2 Redirect

The SSLv2 Redirect feature enables the NetScaler system to redirect any requests from SSLv2 clients to a precise error message, either user configured or internally generated, instructing the client about a further course of action. For example, the redirect could instruct users on how to upgrade

© Copyright 2012 Citrix Systems, Inc.

to an SSLv3 client. The SSLv2 protocol must be disabled in order for SSLv2 Redirect to work; otherwise, the NetScaler system will negotiate the SSLv2 handshake.

SSL Citrix Recommendations

SSL Citrix recommendations include the following:

- Offload SSL processing to an application delivery controller such as NetScaler.
- Be aware of which components in your infrastructure are processing SSL.
- Report on all certificate use and expiration dates.
- Document, measure, and report on SSL performance.
- Test SSL processing with evaluation certificates.
- Use 2048-bit SSL keys.

 © Copyright 2012 Citrix Systems, Inc.

Creating SSL Virtual Servers

An SSL virtual server accepts encrypted traffic, decrypts it, and sends the clear text messages to the services that are bound to the virtual server. This process allows for offloading SSL processing to the NetScaler system and the back-end servers to process a greater number of requests.

Creating an SSL Virtual Server

Create an SSL virtual server in the NetScaler system by specifying the vServer name, service type, IP address, and port of the virtual server under the SSL Node in the configuration utility. If configuring from the command line, specify the following command:

```
add lb vserver <name> <serviceType> [<IPAddress> <port>]
```

At least one service must be bound to a virtual server for the virtual server to accept incoming client requests.

Binding an SSL Virtual Server

At least one service must be bound to a virtual server for the virtual server to accept incoming client requests.

Bind a service to the SSL virtual server by specifying the virtual server name and the service name under the SSL node in the configuration utility. If you are using the command-line interface, enter:

```
bind lb vserver <name> <serviceName>
```

Cipher Suite Binding to an SSL Virtual Server

A cipher group is a set of cipher suites that you bind to an SSL virtual server or service on the NetScaler. A cipher suite consists of a protocol, a key exchange (Kx) algorithm, an authentication (Au) algorithm, an encryption (Enc) algorithm, and a message authentication code (Mac) algorithm. NetScaler contains a predefined set of cipher groups. When you create an SSL Virtual Server, the default cipher group is automatically bound to it.

In addition, you can create a user-defined cipher group and bind it to an SSL virtual server. To create a user-defined cipher group, select one or more groups from the available cipher groups. If you specify a cipher alias or a cipher group, all the ciphers in the cipher alias or group are added to the user-defined cipher group. You can also add individual ciphers to a user-defined group. However, you cannot modify a predefined cipher group.

A user-defined cipher group is configured in the **SSL > Cipher Groups** menu in the configuration utility. An example of a cipher suite algorithm includes the following:

© Copyright 2012 Citrix Systems, Inc.

```
Cipher Name: SSL3-RC4-
SHA Description: SSLv3 Kx=RSA Au=RSA Enc=RC4(128) Mac=SHA1
```

You can bind a cipher group to an SSL virtual server, service, or service group in the **SSL Offload** > **Virtual Servers, Services, or Service Groups** menu in the configuration utility.

For more information about binding cipher groups to virtual servers, see Citrix eDocs at *http://edocs.citrix.com*.

 © Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

1. Which of the following is not a requirement necessary for a front-end SSL_TCP with a back-end TCP configuration?
    a. An installed certkey
    b. An SSL_Bridge configuration
    c. A load-balancing virtual server using the SSL_TCP protocol
    d. A TCP service or services associated with back-end web servers
2. Which of the following does Citrix recommend for SSL?
    a. Disable SSL
    b. Use 1024-bit SSL keys
    c. Test SSL processing with certificates in production
    d. Offload SSL processing to an application delivery controller such as NetScaler
3. To which of the following items can you NOT bind a cipher group to in the SSL Offload menu of the configuration utility?
    a. Service
    b. Back-end server
    c. Service group
    d. SSL virtual server

© Copyright 2012 Citrix Systems, Inc.

Authentication Setup

NetScaler can authenticate users against a variety of sources and can function as a SAML service provider. For example, NetScaler supports Active Directory, RADIUS, LDAP, TACACS, and client side Kerberos authentication, and also can authenticate users with PKI certificates. For more information about setting up Active Directory authentication with NetScaler, see Citrix article CTX111079 at *http://support.citrix.com*. NetScaler also supports authentication for GUVCLI and for Citrix Access Gateway Enterprise Edition (AGEE)/AAA-TM.

The choice of security negotiated between client and server affects:

- The level of authentication protocol used by clients
- The level of session security negotiated
- The level of authentication accepted by servers

For Security Assertion Markup Language (SAML), NetScaler accepts tokens for version 2.0, enabling you to configure SSO on the NetScaler for users who log on through a third-party authentication server that supports SAML. As a service provider in SAML, NetScaler supports both authentication models—service provider-initiated and IdP (Identify Providers)-initiated. Supported IdP for NetScaler 10 include Secureauth, Shibboleth, ADFS, and Cloud Gateway.

AAA for Traffic Management

The AAA-TM (Authentication, Authorization, and Accounting for Traffic Management) feature of NetScaler fully supports NTMLv2 (Windows NT LAN Manager version 2) single sign-on (SSO) and resolving issues with accessing Microsoft SharePoint servers that are part of a private network behind an AAA-TM SSO configuration.

The NetScaler system attempts to connect by using NTMLv2 first, and only if it fails does it then attempt to fall back to NTMLv1. NetScaler also supports NTLMv2 session security and signing support for type 3 messages.

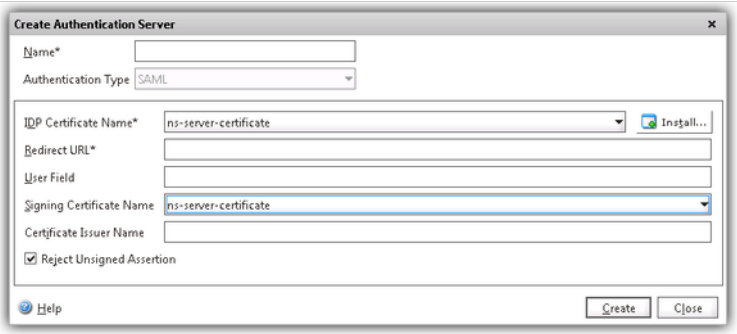
SAML 2.0 Consumer Support

For Security Assertion Markup Language (SAML), NetScaler accepts tokens for version 2.0, enabling you to configure SSO on the NetScaler for users who log on through a third-party authentication server that supports SAML. As a service provider in SAML, NetScaler supports both authentication models-service provider-initiated and IdP (Identify Providers)-initiated. Supported IdP for NetScaler 10 include Secureauth, Shibboleth, ADFS, and Cloud Gateway.

To configure this feature using the NetScaler command-line interface, type the following command:

 © Copyright 2012 Citrix Systems, Inc.

```
add authentication samlAction <name> [-samlIdPCertName <string>]
[-samlSigningCertName <string>][-samlRedirectUrl <string>][-
samlUserField<string>][-
samlRejectUnsignedAssertion ( ON | OFF )] [-
samlIssuerName <string>]
```



Authentication Policy Creation

NetScaler supports LDAP, RADIUS, TACACS+, and NT4 authentication. To configure authentication, you must create authentication policies. You can configure one or more authentication policies depending on your authentication needs. Authentication policies, similar to other policies for NetScaler, consist of an expression and an action.

For information about how to create a policy, see Citrix eDocs at *http://edocs.citrix.com*.

Authentication Policy Binding

After creating an authentication policy, you bind it to the system global entity and assign it a priority. You can create simple server configurations by binding a single authentication policy to the system global entity, or you can bind multiple policies to the system global entity. If no authentication policies are bound to the system, users are authenticated by the onboard system.

User accounts must be configured on the NetScaler before users can be externally authenticated. For more information about authentication policy binding, see Citrix eDocs at *http://edocs.citrix.com*.

For step-by-step instructions about binding an authentication policy globally using the configuration utility, see Citrix eDocs.

 © Copyright 2012 Citrix Systems, Inc.

Resource Identification for Users

Authorization policies are used to provide users with access to specific resources. There are only two actions for authorization policies: ALLOW and DENY. The default setting for authorization when no specific policy exists is to deny access to network resources. Citrix recommends to create an authorization policy that explicitly allows a user or group of users to access the resources that they require.

© Copyright 2012 Citrix Systems, Inc.

Single Sign-on Setup

After configuring and binding authentication and authorization policies for users, you can customize additional settings for users. For example, enabling or disabling the single sign-on (SSO) setting determines whether the NetScaler will log users on all web applications automatically after they authenticate, or will pass users to the web application logon page to authenticate for each application.

To configure this action, you can create a profile for each user or user group and configure custom session settings. Another option is to create policies to select which connections apply to particular profiles and bind policies to users or groups. You can also bind a policy to the authentication virtual server that handles the traffic to which you want to apply the profile. For more information about the single sign-on setting, see Citrix eDocs at *http://edocs.citrix.com* and Citrix Knowledge Base article CTX132366 at *http://support.citrix.com*.

 © Copyright 2012 Citrix Systems, Inc.

LDAP Authentication

LDAP authentication requires identical group names in Active Directory, on the LDAP server, and on the NetScaler. The characters and case must also be the same. If the root directory of the LDAP server is specified, all of the subdirectories are also searched to find the user attribute. In large directories, this can affect performance. For this reason, Citrix recommends that you use a specific organizational unit (OU).

To configure LDAP authentication, you create an authentication policy and specify LDAP for the authentication type before providing server, IP address, port, and settings information. After the LDAP server settings are configured on the NetScaler, bind the policy to the system global entity. For more information about configuring LDAP authentication, see Citrix eDocs at *http://edocs.citrix.com*.

NetScaler can also query LDAP groups and extract user information from ancestor groups configured on the same authentication server. This LDAP-nested group extraction is configured using an authorization policy. For more information about LDAP-nested group extraction including the search filter syntax, see Citrix eDocs at *http://edocs.citrix.com* and Citrix article CTX123795 at *http://support.citrix.com*.

Addtionally, you can configure NetScaler to extract an Active Directory group and enable clients to access the NetScaler VPN based on the Active Directory groups by using the LDAP authentication and the Groups Allowed to Login feature. For more information about Active Directory group extraction for LDAP, see Citrix article CTX125797 at *http://support.citrix.com*.

© Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

1. True or False: To configure authentication on a NetScaler you must create authentication policies.
    a. True
    b. False
2. What action should be taken after creating an authentication policy?
    a. Configure and apply the policy.
    b. Bind the policy to a web server.
    c. Prioritize the policy with other authentication policies.
    d. Bind the policy to the system global entity and assign a priority.
3. True or False: NetScaler can query LDAP groups and extract user information from ancestor groups configured on the same authentication server.
    a. True
    b. False

 © Copyright 2012 Citrix Systems, Inc.

Access Control Lists

Access control lists can be configured on the NetScaler system. The NetScaler system compares incoming packets against the access control lists. If a packet matches an access control list rule, the action specified in the rule is applied to the packet.

All access control lists are enabled by default. The NetScaler system compares incoming packets against all the access control lists. If an access control list is not required as part of the lookup table but is retained in the configuration, then it should be disabled before applying the access control lists. Once the access control lists have been applied, the system does not compare incoming packets against disabled access control lists.

Simple Access Control Lists

You can configure simple access control list packet filters based on only the source IP address and the port. If both simple and standard access control lists are configured, simple access control lists are applied first. If the simple access control list returns a DENY, then the configured action is taken. If the simple access control list returns an ALLOW, then the standard access control list is applied, and the return value determines the action. The NetScaler system allows 100,000 simple access control lists to be configured.

The standard access control list limit is 1024, but simple access control lists can be created as memory allows.

Access Control Lists Precedence

Access control list precedence is determined according to the following rules:

- The most specific access control list takes priority unless the priority is assigned.
- The destination takes priority over the source when no priority is assigned.
- The first access control list that matches determines how the packet is processed. The packet is only affected by a single access control list or no access control lists if there is no match.

Matching Access Control List Entries

The NetScaler system can implement IP address-based traffic control on data that it handles using access control lists. This functionality is part of the NetScaler system, and it does not need to be enabled.

The ways in which traffic can be handled once it is matched are:

If no access control list entry is matched, then the traffic is allowed.

Access Control List Configuration

You can perform the following access-control-list-related processes on the NetScaler system:

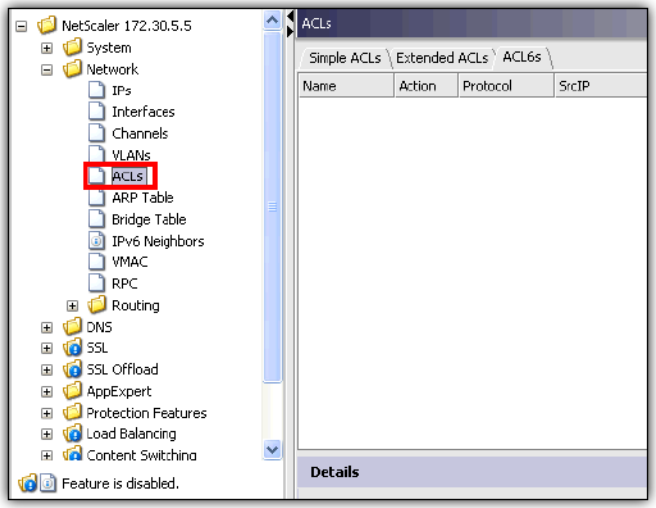
- Create access control lists
- Modify access control lists
- Define access control list processing mode
- Define access control list priority
- Activate access control lists
- Disable access control lists
- Remove access control lists

Access control list entries can be viewed in the configuration utility under the Networking node and under the access control list subnode. New access control list entries can be added, modified, and removed.

Access control lists are used to restrict access to the configuration utility from specific IP addresses or ranges. Access control lists can be used to force configuration utility connections to be allowed only over secure ports and HTTPS instead of unsecured ports, such as HTTP.

By default, access control lists are assigned priorities in the order in which they are created if an explicit priority is not specified. The system defaults to assigning priorities by multiples of 10 to provide spacing between access control lists. Explicit priorities are assigned to access control lists using the add command when creating an access control list or when using a set command to modify an access control list. These commands allow you to configure the order in which access control lists are processed.

 © Copyright 2012 Citrix Systems, Inc.



Simple Access-Control-List Configuration (Command-line Interface)

You can use the following command to create a named access-control-list entry using a source IP address and destination IP address (`add acl` or `add ns acl` is allowed):

```
add ns acl aclNameaclAction -srcIP IPAddress -destIP IPAddress
```

Command-line Interface

```
add ns acl test1 ALLOW -srcIP 10.10.1.1-10.10.1.3 -
destIP 10.10.1.4
```

Detailed Access Control Lists Configuration

You can use the following command to add an access control list to the system configuration in the command-line interface:

```
add ns acl aclNameaclAction
-srcIP operatorsrcIPVal
-srcPort operatorsrcPortVal
-destIP operatordestIPVal
-destPort operatordestPortVal
-TTL PositiveInteger -srcMac MACAddress
-protocol protocol -established -vlan PositiveInteger
-interface InterfaceName -icmpType PositiveInteger
-icmpCode PositiveInteger -priority PositiveInteger
-state ENABLED
```

Each inbound packet is matched against configured access control lists, and the specified action is applied to the packet. This entry can be enabled or disabled as needed. The `apply ns acls` command takes all enabled access-control-list entries and makes them active on the NetScaler system.

Extended Access Control Lists

Extended access control lists (ACLs) filter data packets on the basis of various parameters, such as source IP address, source port, action, and protocol. An extended ACL defines the conditions that a packet must satisfy for the NetScaler to process the packet, bridge the packet, or drop the packet.

If both simple and extended ACLs are configured, incoming packets are compared to the simple ACL first. If the packet does not match any simple ACL, the NetScaler compares the packet characteristics to those specified in any configured extended ACLs. If the packet matches an extended ACL, the NetScaler applies the action specified in the extended ACL. NetScaler supports up to 10,000 extended ACLs.

Creation and Removal of Access Control Lists

You can use the following command to enable access-control-list entries in the command-line interface:

```
add ns acl aclName aclAction
```

This command creates the named access control list on the system. The access control list is enabled but not activated (applied) by default. You must run the `apply ns acls` command after the `add ns acl` command in order to activate the access control list.

 © Copyright 2012 Citrix Systems, Inc.

The `remove ns acl` command is used to remove individual access control lists from the system. However, the `apply ns acls` command should be used to apply the change to the NetScaler system. You can use the following command to remove an access control list in the command-line interface:

```
remove ns acl aclName
```

To remove all access control lists from the system, or clear all access control lists in the command-line interface, you can use the `clear ns acls` command. This operation does not require an explicit application.

Access Control Lists Application

You can use the following command to commit the access control list to the configuration and to activate the access control list on the system in the command-line interface:

```
apply ns acls
```

This command is required after any modification is made to the access control list; it activates access control lists and commits changes to the system. After running this command, all existing sessions are re-validated and any sessions matching a drop command are dropped.

Each access control list has an active status and an applied status. The active status indicates whether an access control list is ENABLED or DISABLED. The applied status indicates whether the access control list is APPLIED or NOTAPPLIED. Applied access control lists are saved to the configuration, and the active status determines whether traffic is compared against the access control list. However, if an access control list is part of the running configuration, it will be saved, regardless of applied status.

Any change to an access control list requires that the `apply ns acls` command is issued to propagate the change to the configuration. The `apply ns acls` command allows you to change the access-control-list configuration without changes taking effect until the apply command is issued. If access control lists are selectively disabled or enabled, then you should use the `enable ns acl` command and `disable ns acl` commands. Disabled access control lists are applied but not active. They are retained by the system following a `save ns config` command.

You can use the following commands to commit a change to an access control list after running the apply ns acls command in the command-line interface:

```
set ns acl
enable ns acl
disable ns acl
```

Viewing of Access Control Lists

You can use the following command to display the access control lists in the command-line interface:

```
show ns acl [aclName]
```

This command shows access control lists on the NetScaler system and the configuration details, including active and applied status information. If a name is specified, then only that particular access-control-list information is displayed. If it is not specified, all configured access control lists are displayed.

You can use the following command to display access-control-list statistics in the command-line interface:

```
stat ns acl aclName -detail -fullValues -ntimes PositiveInteger -
logFile InputFilename
```

Removal of Access Control Lists from the System

The `remove ns acl` command is used to remove individual access control lists from the system. However, the `apply ns acls` command should be used to apply the change to the NetScaler system. You can use the following command to remove an access control list in the command-line interface:

```
remove ns acl aclName
```

To remove all access control lists from the system, you can use the `clear ns acls` command. This operation does not require an explicit application.

Access-Control-List Example

The following example demonstrates how to configure an SSH access control list in the command-line interface. The following values are used in the example:

- NSIP address: 192.168.0.2, 192.168.0.3
- MIP address: 192.168.0.1
- Trusted Network A: 10.10.1.1-20

The following command example shows configuring SSH access control list in the command-line interface:

 © Copyright 2012 Citrix Systems, Inc.

```
add acl bastion1-ssh ALLOW -srcip 10.0.0.1-10.0.0.20 -
destip 192.168.0.1-192.168.0.3 -dstport 22
-protocol TCP
```

The bastion1-ssh access control list is used to create an access control list to allow traffic from a trusted network to connect to the NetScaler-owned IP addresses for the high-availability pair over port SSH. This example secures access to the command-line interface from an allowed range of trusted bastion hosts.

IPv6-based Access-Control-List Support

Access control lists based on Internet Protocol version 6 (IPv6) are configured in the configuration utility. To create an IPv6 access control list, create an extended access control list and select the appropriate IPv6 protocol.

The NetScaler system extends its IPv6 support to server-side implementation. End-to-end IPv6 support enables the use of IPv6 addresses for SNIP addresses, virtual servers, services, and servers. IPv6 management utilities can also be used, such as Ping6 and Traceroute6. IPv6 support also extends to OSPFv3.

Static routes can be:

- Configured using IPv6 addresses to any destination
- Assigned values for distance and cost
- Enabled in the advertising of static routes to IPv6 routing protocols

Core Protection Enhancements

NetScaler contains a core-protection layer that protects and secures infrastructures, networks, and applications. NetScaler provides security from layer-2 to layer-7 of the OSI model, including protection against various kinds of Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks, TCP SYN flood attacks, and traffic surges.

In NetScaler 10, the following core protection enhancements are included:

- SYN cookie protection
- Runtime detection of SYN attacks
- Protection against TLS Renegotiation attacks
- Adaptive Request Timeout for HTTP DoS

 © Copyright 2012 Citrix Systems, Inc.

SYN Protection

NetScaler often resides on the edge of the network and encounters many external attacks. Because of this, NetScaler contains an array of security features designed to protect against unauthorized attacks, including distributed denial of service (DD-S) attacks that can cripple a system with a barrage of time- and resource-draining flows including slow rate attacks that attempt to avoid detection.

For SYN flood attacks, NetScaler provides superior protection by implementing an enhanced SYN cookie mechanism that operates at wire-speed, enabling it to protect against denial of service (DoS) attacks.

To mount a SYN flood attack, a hacker initiates a large number of TCP connections but does not respond to the SYN messages sent by the targeted server. The source IP addresses in the SYN messages received by the server are typically fake. Because new SYN messages arrive before the partially open connections initiated by previous SYN messages time out, the number of connections increase until the server does not have enough memory to accept new connections.

NetScaler protects against these types of SYN attacks with a runtime-detection engine. With SYN denial of service (DoS) protection, the memory of NetScaler is not wasted on false SYN packets and is only used to serve legitimate clients. Normal TCP communications with legitimate clients continue uninterrupted, even when the web site is under a SYN flood attack.

NetScaler currently performs SYN cookie mechanisms by default for all TCP traffic. For the TCP applications like SMTP, where the server initiates traffic, if a FINAL-ACK message from the client is dropped, then the client connection hangs. To avoid this, SYN protection can also be turned off if necessary.

© Copyright 2012 Citrix Systems, Inc.

SSL Renegotiation Attack

The SSL and TLS renegotiation process is vulnerable to an attack in which the attacker forms a TLS connection with the target server, injects content of its choice, and then splices in a new TLS connection from a client. NetScaler provides protection against these attacks by cryptographically binding renegotiation handshakes to the enclosing TLS cryptographic parameters, thus allowing the server to differentiate renegotiation from initial negotiation, as well as preventing a renegotiation from being spliced in between connections.

For more information about SSL renegotiation, see Citrix article CTX121925 at *http://support.citrix.com*.

 © Copyright 2012 Citrix Systems, Inc.

TLS Man-in-the-Middle Attacks

NetScaler 10 mitigates TLS man-in-the-middle attacks through the inclusion of support for TLS renegotiation extensions as described in the Internet Engineering Task Force (IETF) standard RFC5746. This protection is only included for MPX and VPX systems.

For more information, see *http://tools.ietf.org/html/rfc5746* and Citrix article CTX123680 at *http://support.citrix.com*.

Adaptive Request Timeout for HTTP DoS

HTTP denial-of-service (DoS) protection provides an effective way to prevent attacks from reaching your protected web servers. The HTTP DoS feature also ensures that a NetScaler located between the cloud and your web servers is not disabled by an HTTP DoS attack.

When the NetScaler detects an attack, it responds to a percentage of incoming requests with a Java or HTML script containing a simple refresh and cookie. Legitimate web browsers and other web-based client programs can parse this response and then resend a "POST" request with the cookie. DoS clients drop the NetScaler system response instead of parsing it, and their requests are therefore dropped as well.

To counter slow HTTP DoS attacks, NetScaler employs Adaptive Request Timeout, taking into consideration factors like payload length, MSS, inter-packet delay, and RTT to adapt the timeout used for requests. When the timeout occurs, one of the possible configured actions is taken—"DROP" or "RESET".

 © Copyright 2012 Citrix Systems, Inc.

Slow-Header, Slow-Post, and Slow-Read Attacks

While many networking systems have inherent protection against DoS attacks, a new class of attacks has shown the ability to circumvent the inherent protection offered in these systems. These attacks use a common form of the HTTP method used in most applications, thus causing resource consumption on the server. The attacks are carefully crafted and affect many networking systems that perform HTTP processing. The impact is greater on HTTP proxies, which try to simulate the HTTP stack while managing the end-to-end request and response flow.

Slow-Header Attack

In a slow-header attack, the attack finds the approximate timeout value set on the server side and then chooses a client value which is lower than the configured value. The attack then initiates an HTTP request with a partial header to the server, and keeps sending one header based on the chosen value. The timeout is not triggered on the server, the request cannot be completed, and the connection is dropped when the client is found idle during the time interval. NetScaler has built-in protection against slow-header attacks, which is enabled by default. Connections that send partial headers after the client idle timeout are flushed silently. NetScaler studies the flow of traffic, identifies the slow header connection, and drops them when time expires. NetScaler counters also support the tracking of these connections.

Slow-Post Attack

In a slow-post attack, an HTTP post request with large content-length header value is sent, which causes the server or proxies to believe that the client is going to send the data specified in the header. The server keeps the connection open to receive the data, but the attack client sends one byte of "POST" data at a regular time interval configured by the attacker, so the connection remains alive. Since the connection is alive, the client idle timeout is not triggered and the server keeps the connection alive until all the bytes of data specified in the header are received by the server, which can reduce server performance. NetScaler can protect against this attack if the request timeout parameter is configured. Any transaction which is not completed during the time interval is dropped and the client is considered an unauthorized client. The `Show connectiontable` command can be used to verify that these connections are cleaned up.

Slow-Read Attack

In a slow-read attack, the client establishes a connection to the server and sends an appropriate HTTP request. However, the client reads the response at a very slow speed. Some slow-read-attack clients do not read the response for a long time, and then start reading the data one byte at a time just before the idle connection timeout. The client sends a "Zero" window to the server, causing the server to assume that the client is busy reading the data. As a result, the server keeps the connection open for an extended period of time. Multiple slow-read connections to the server

© Copyright 2012 Citrix Systems, Inc.

consume the resources and can cause it to be unresponsive to new requests. NetScaler protects against slow-read attacks by automatically identifying TCP segments sent from a client with a window value less than 1 MSS (maximum segment size). If idle for a predefined set of time, NetScaler silently drops the connections. If a malicious client sends a window of 1 MSS or greater and reads the response data a very slow speed, it is treated as a legitimate client, but NetScaler can be configured to drop such packets as part of its attack-protection mechanism.

 © Copyright 2012 Citrix Systems, Inc.

Client Traffic Management Based on Traffic Rate

You can monitor the rate of traffic that flows through load-balancing virtual servers and control the behavior of the NetScaler system based on the traffic rate. You can throttle the traffic flow if it is too high, cache information based on the traffic rate, and if the traffic rate is too high, redirect excess traffic to a different load-balancing virtual server.

You also can apply rate-based monitoring to HTTP and DNS requests. These methods can be used to prevent brute force attacks and "site-scraping." For more information about configuring rate-based policies, see Citrix article CTX132362 at *http://support.citrix.com*.

Test Your Knowledge

1. Which is NOT affected by the choice of security negotiated between the client and server?
    a. The level of session security negotiated
    b. The level of authentication accepted by servers
    c. The level of authentication protocol used by clients
    d. Levels of session security that have yet to be negotiated
2. If both simple and extended ACLs are configured, how are incoming packets handled?
    a. All incoming packets are denied.
    b. All incoming packets are accepted.
    c. Incoming packets are compared to the extended ACLs, which override any simple ACLs configured.
    d. Incoming packets are compared to the simple ACLs first. If the packet does not match any simple ACLs, the NetScaler compares the packet characteristics to those specified in any configured, extended ACLs.
3. True or False: NetScaler provides protection against SSL renegotiation attacks by cryptographically binding renegotiation handshakes to the enclosing TLS cryptographic parameters.
    a. True
    b. False

 © Copyright 2012 Citrix Systems, Inc.

Module 11

Responder, Rewrite, and URL Transform

© Copyright 2012 Citrix Systems, Inc.

Overview

The responder feature bases responses on who sends the request, where it is sent from, and other criteria with security and system management implications. The rewrite feature rewrites information in the requests or responses handled by the NetScaler system. The URL transformation feature provides a method for modifying all URLs in designated requests from an external version seen by outside users to an internal version seen only by your web servers and IT staff.

After completing this module, you will be able to:

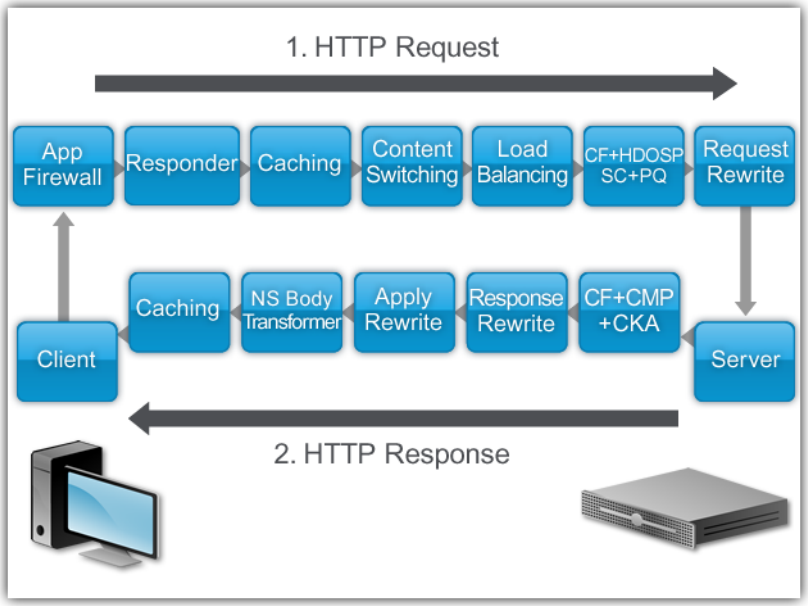
- Identify recommendations and uses for advanced policy expressions.
- Configure rewrite actions and policies.
- Configure NetScaler system options to protect against unauthorized attacks.
- Configure responses to specific requests.
- Configure a URL transformation policy.

© Copyright 2012 Citrix Systems, Inc.

Understanding Packet-Processing Flow

As traffic through the NetScaler system is evaluated by the different features, each feature performs policy evaluation. Whenever a policy matches the traffic, the NetScaler system stores the action and continues processing. The NetScaler system typically applies all matching actions after processing is completed.

This figure illustrates the overall packet-processing flow for evaluation order. For the majority of the features, the packet evaluation order does not matter because the events are logged. The important factor is that some of the features, such as application firewall, responder, and caching, are first in the packet-processing flow. If it matches these, it may not continue evaluation depending on the configured action. These specific feature evaluations are performed more frequently on the incoming request, not the transformed data.



Legends for acronyms used in image above:

Acronym	Phrase
CF	Content filtering
HDOSP	HTTP denial of service protection
SC	Secure connection
PQ	Priority queueing

 © Copyright 2012 Citrix Systems, Inc.

Acronym	Phrase
CMP	Compression
CKA	Client keep-alive mode

Understanding Policies

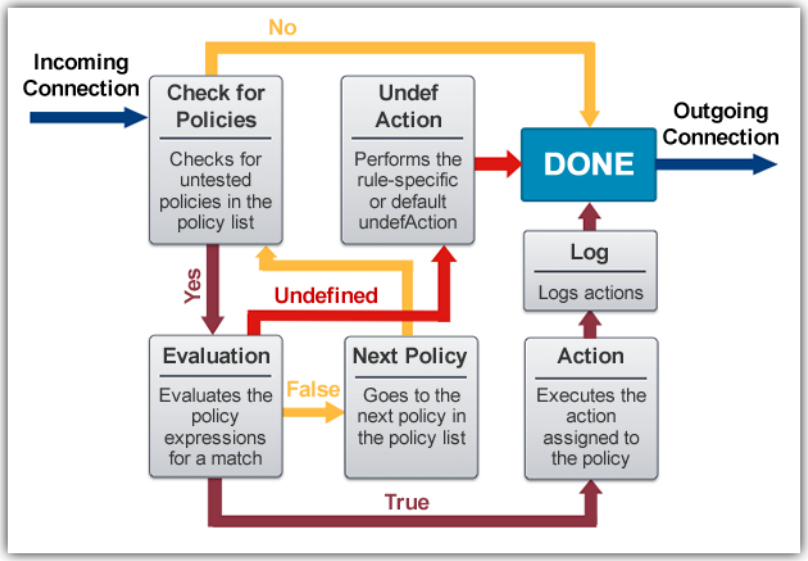
The NetScaler system uses the policies to evaluate specified conditions and to define the actions to be taken if conditions are met. The actions defined are specific to the feature for which the policy is created. The order and flow of policy evaluation depends on the feature set and policy-expression type.

Policy Expressions

Policy expressions compare information to a particular value to determine whether the connection matches the policy. The end result of any complete policy expression must produce a Boolean value or, with advanced policy expressions, an UNDEF value, to tell the NetScaler system explicitly whether a particular connection matches the policy. When a policy expression is bound or activated, the feature set being bound to or activated evaluates whether the policy expression is valid for that specific feature.

 © Copyright 2012 Citrix Systems, Inc.

Policy-Process-Evaluation Flow



As the figure shows, the evaluation of a policy produces three possible outcomes:

True	A true outcome means that the connection matches the policy expression.
False	A false outcome means that the connection did not match the policy expression.
Undefined	An undefined outcome occurs when a policy cannot be evaluated.

The exact packet-processing-evaluation flow depends on the feature set performing the evaluation.

Identifying Advanced Policy Expressions

The advanced-policy-expression language is an object-oriented, general purpose expression language capable of extremely precise data manipulation.

Advanced-policy-expression types include:

- Non-compound advanced policy expressions
- Compound advanced policy expressions

In a policy, advanced policy expressions can be combined using Boolean operators, allowing the NetScaler system to test multiple connection elements in a structured manner. Advanced expressions are supported in several features, including DNS, GSLB, content switching, responder, and rewrite policies.

Advanced-Policy-Expression Syntax

NetScaler advanced policy expressions read from left to right. The left-most term designates what part of the connection the expression is analyzing. The values described in the following list can appear in this position.

CLIENT	Designates the client portion of the connection. Use this term to extract information from the client, such as the client IP address.
HTTP	Designates a complete HTTP connection, including both the request and the response. Use this term to extract information from an HTTP connection.
SERVER	Designates the server portion of the connection. Use this term to operate on information from the server, such as the hostname.
SYS	Designates the NetScaler system and NetScaler operating system. Use this term to operate on the date and time of the connection or to insert a classic expression in a larger advanced policy expression.
TARGET	Designates the connection target. Use this term to operate on the target.

 © Copyright 2012 Citrix Systems, Inc.

TEXT	Designates any piece of text. Use this term to operate on any text values associated with the connection.
URL	Designates the URL in the HTTP request. Use this term to operate on the URL.
VPN	Designates any connection that occurs through a virtual private network VPN that is managed by the NetScaler system.

Successive terms to the right refine the first term to define specific attributes of the connection. Each term is separated from the preceding or the following terms with a period. Arguments appear in parentheses following the term to which they apply.

Actions

An action:

- Is bound to or activated by policies
- Cannot depend on results of other actions
- Is applied at the end of the policy evaluation process
- Is owned by individual NetScaler features

  For example, actions configured in the responder module are different from those configured in the rewrite module. The individual feature ensures that the respective actions are applied.

Action Syntax

At the NetScaler command prompt, type the following commands to add the rewrite action command:

```
add rewrite action name type target [stringBuilderExpr] [(-pattern
expression]
```

Example of NetScaler Command Prompt to Add a Rewrite Action

```
add rewrite action Replace_HTTP_to_HTTPS INSERT_AFTER
"HTTP.RES.HEADER(\"Location\").Value(0).Prefix(4)" "\"s\""
```

 © Copyright 2012 Citrix Systems, Inc.

Understanding Bind Points

You can bind a classic compression policy either at the global level or at the virtual server (content-switching or load-balancing virtual server) level. You can also bind a given compression policy to more than one bind point. The priority values assigned to a policy that is bound to various bind points need not have the same value. For example, if you bind a policy both globally and to a load-balancing virtual server, you can assign the policy a priority of 10 at the global bind point and a priority of 100 at the virtual server bind point. Therefore, the priorities assigned to the classic compression policies that are bound to various bind points at any given time might be an assortment of custom priority values and default values.

If you do not assign a policy a priority value, the NetScaler system assigns the classic compression policy a default priority value of zero.

The NetScaler system evaluates classic policies based first on assigned priorities, then on bind points, and finally on the chronological order in which the policies were configured.

During classic policy evaluation, priority values are considered first. The NetScaler system begins with the policy that has the lowest priority value regardless of the bind point to which the policy is bound. Then, the system evaluates policies in ascending order of priority values while moving from one bind point to the other if necessary.

If multiple policies bound to different bind points have the same priority value, the bind points of those policies are considered, with the order of evaluation progressing from the most specific bind point to the least specific bind point. For example, if classic policies are bound globally to a content-switching virtual server and to a load-balancing virtual server, the policies bound to the load-balancing virtual server are evaluated first because, in this configuration, the load-balancing virtual server represents the most specific bind point. The NetScaler system then evaluates the policies that are bound to the content-switching virtual server. Finally, the system evaluates the policies that are bound at the global level.

If two or more policies have the same priority value, the chronological order in which the policies were configured is considered. The policy that was configured first is evaluated first, followed by the policy that was configured second, and so on.

Policy Binding Evaluation Process

In the policy evaluation process, a request is being evaluated by the NetScaler system. The following explains the evaluation process:

1. The bind point selector selects a bank based on the configuration.
2. The NetScaler system evaluates the request for matches in the selected bank.
3. The NetScaler system finds a match and the policy evaluation process ends. The specified actions are performed.
4. The NetScaler system invokes the next bank of policies and the request is evaluated against the next policy bank if no match is found.

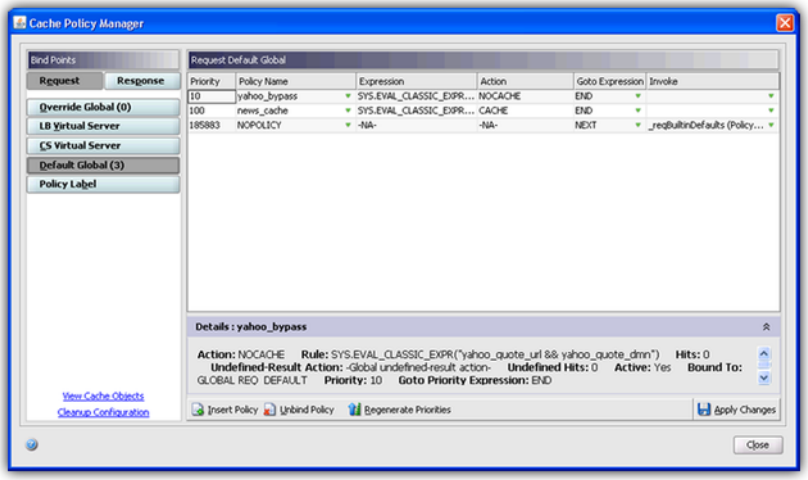
5. The NetScaler system performs the identified action if a match is found.

   The NetScaler system performs the rule-specific or the global UndefAction in instances when no match is found and all of the identified policy banks and policies have been evaluated.

Binding in the Policy Manager

The policy manager provides an easy interface for managing bind points, priorities, and policy labels in the configuration utility. The policy manager makes configuring priorities within bind points much easier because the logic is built-in and will not let you configure faulty logic. For example, the GoToExp logic is built into the policy manager. The NetScaler system cannot use GoToExp to go to another policy label. The first three policy labels are most frequently used. Configuring for a global bind point results in all traffic being evaluated.

The virtual server bind point applies to a subset of the global traffic. For example, content switching will hit a subset of the traffic through that virtual server. When binding, start with the most specific type first, then go to the content-switching virtual server, and then to the global default. Lastly, there is a special policy label that can global override all of the prior three labels. The processing is the same as it is listed in the policy manager dialog box in the configuration utility.



 © Copyright 2012 Citrix Systems, Inc.

Understanding Policy Labels

A policy label is an end-user-defined bind point to which policies are bound. The `policy label` command allows you to logically group policies and define the order in which they are evaluated. The policy labels must be invoked to evaluate the end-user-defined policy banks.

Policy labels are invoked from other policies. When a policy label is invoked, all the policies bound to it are evaluated in the order of priority. When a policy is matched and the expression is evaluated to be true, the appropriate action is logged, and the control is returned to the policy that invoked the policy label.

The policy defines the next policy to be evaluated using GoToExp, or the action ends and goes to the next priority policy or the next policy bank. Policy labels are specific to certain features, such as the rewrite and responder features.

Configuring Policy Labels

At the NetScaler command prompt, type the following commands to add a rewrite policy label and responder policy label:

```
add rewrite policylabel policyLabelName
http_req|http_res|url|text|clientless_vpn_req|clientless_vpn_res
```

```
add responder policylabel policyLabelName
```

Example of NetScaler Command Prompt to Create a Rewrite Policy Label

```
add rewrite policylabel lbl-respndr-pol
```

Example of NetScaler Command Prompt to Create a Responder Policy Label

```
add responder policylabel lbl-respndr-pol http_req
```

Pattern Sets

A pattern set is an array of indexed patterns that you configure on the NetScaler system. Pattern sets are used for string matching during advanced policy evaluation. The NetScaler system provides you with a set of advanced expression operators that you can use to compare a string in a packet with the patterns that are indexed and stored in a pattern set.

You can configure the system to compare the string that is identified in a packet with one or more patterns in the pattern set by using a simple advanced expression combined with an operator. Additionally, after you create a pattern set, you can use the pattern set in multiple advanced policies. Therefore, pattern sets eliminate the need for you to configure compound advanced expressions that perform string matching with multiple OR operations (one expression for each comparison). This also reduces the consumption of system resources in terms of memory and the number of expressions that the system has to evaluate.

First, you create a pattern set and bind patterns to it. Then, when you configure a policy for comparing a string in a packet with the pattern set, you use an appropriate operator and pass the name of the pattern set as an argument.

How String Matching with a Pattern Set Works

A pattern set contains a set of patterns, and each pattern is assigned a unique index. During policy evaluation, the operator compares the string that is identified in the packet with the patterns defined in the pattern set until a match is found. Then, depending on its function, the operator returns either a Boolean value that indicates whether a matching pattern was found or the index of the pattern that matches the string.

Configuring a Pattern Set

You configure a pattern set by specifying the strings that are to serve as patterns and binding them to the pattern set. Each pattern in the set has a unique index value. If you specify an index for the first pattern that you bind to a pattern set, you must specify an index for all the other patterns in the set, and you can modify the values at any time. If you do not specify an index for the first pattern, the NetScaler system assigns the pattern an index of value 1. Thereafter, the system assigns an index value to all the patterns that you bind to the set, and you cannot change them. After you have configured a pattern set for which the system has generated index values automatically, if you want to assign index values of your choice, you must create a new pattern set.

Index values are not regenerated automatically if one or more patterns are deleted or modified. For example, if the set contains five patterns, with indexes from 1 through 5, and if the pattern with an index of 3 is deleted, the other index values in the pattern set are not automatically regenerated to produce values from 1 through 4.

In the NetScaler command-line interface, you first create a pattern set by using the `add policy patset` command. Then, you create patterns and bind them to the pattern set, one pattern at a

 © Copyright 2012 Citrix Systems, Inc.

time, by using the `bind policy patset` command. In the NetScaler configuration utility, you perform all these tasks in a single dialog box.

> Pattern sets are case sensitive. Therefore, the string pattern "product1," for example, is not the same as the string pattern "Product1."

For more information about configuring a pattern set, see Citrix eDocs at *http://edocs.citrix.com*.

Using Pattern Sets

After you configure a pattern set, you can use it in an advanced expression that passes the pattern set as an argument to an appropriate operator.

Following is the format of the default syntax expression that compares a string with a pattern set:

```
<text>.<pattern set operator>("<name>")
```

In the preceding expression format, <text> represents any default syntax expression that identifies a string in a packet and <name> is the name of the pattern set.

> If you want the system to determine whether the value of the host header in a client request contains any of the patterns that are configured in a pattern set called "Patternset1," you can use the following default syntax expression:
> `HTTP.REQ.HEADER("Host").CONTAINS_ANY("Patternset1")`

Pattern Set Operators

The following table describes the operators that you can use with pattern sets. When you use an operator, replace <text> with the default syntax expression that identifies the string with which you want to perform string matching, and replace <pattern_set_name> with the name of the pattern set.

Operator	Description
<text>.CONTAINS_ANY(<pattern_set_name>)	Evaluates whether the target text contains any of the patterns that are bound to <pattern_set_name> and returns a Boolean TRUE if one or more matching patterns are found.

Operator	Description
<text>.EQUALS_ANY(<pattern_set_name>)	Evaluates whether the target text exactly matches any of the patterns that are bound to <pattern_set_name> and returns a Boolean TRUE or FALSE to indicate the result of the evaluation.
<text>.ENDSWITH_ANY(<pattern_set_name>)	Evaluates whether the target text ends with any of the patterns that are bound to <pattern_set_name> and returns a Boolean TRUE or FALSE to indicate the result of the evaluation.
<text>.STARTSWITH_ANY(<pattern_set_name>)	Evaluates whether the target text starts with any of the patterns that are bound to <pattern_set_name> and returns a Boolean TRUE or FALSE to indicate the result of the evaluation.
<text>.STARTSWITH_INDEX(<pattern_set_name>)	Evaluates whether the target text starts with any of the patterns that are bound to <pattern_set_name> and, if a match is found, returns the numerical index of the matching pattern.
<text>.ENDSWITH_INDEX(<pattern_set_name>)	Evaluates whether the target text ends with any of the patterns that are bound to <pattern_set_name> and, if a match is found, returns the numerical index of the matching pattern.
<text>.CONTAINS_INDEX(<pattern_set_name>)	Evaluates whether the target text contains any of the patterns that are bound to <pattern_set_name> and, if a match is found, returns the numerical index of the matching pattern.
<text>.EQUALS_INDEX(<pattern_set_name>)	Evaluates whether the target text exactly matches any of the patterns that are bound to <pattern_set_name> and, if an exact match is found, returns the numerical index of the pattern.
<text>.SUBSTR_ANY(<pattern_set_name>)	Selects the first string that matches any pattern in the given pattern set.

 © Copyright 2012 Citrix Systems, Inc.

Typecasting

Typecasting allows you to extract data of one type (such as, text or an integer) from requests and responses and transform it to data of another type. For example, you can extract a string and transform the string to time format. You can also extract a string from an HTTP request body and treat it like an HTTP header or extract a value from one type of request header and insert it in a response header of a different type.

After typecasting the data, you can apply any operation that is appropriate for the new data type. For example, if you typecast text to an HTTP header, you can apply any operation that is applicable to HTTP headers to the returned value.

Typecasting is only available with the advanced policy engine.

The following table describes some of the values for controlling typecast results. For more information about typecasting data including more typecasting functions and examples, see Citrix eDocs at *http://edocs.citrix.com*.

Function	Description
<text>.TYPECAST_LIST_T(<separator>)	Treats the text in an HTTP request or response body as a list whose elements are delimited by the character in the <separator> argument. Index values in the list that is created start with 0. Text mode settings have no effect on the separator. For example, even if you set the text mode to IGNORECASE, and the separator is the letter “p,” an uppercase “P” is not treated as a separator.
<text>.TYPECAST_TIME_T	Treats the designated text as a date string. The following formats are supported: • RFC822: Sun, 06 Nov 1994 08:49:37 GMT • RFC850: Sunday, 06-Nov-94 08:49:37 GMT • HTTP Set-Cookie Expiry date: Sun, 06-Nov-1994 08:49:37 GMT
<text>.TYPECAST_HTTP_ URL_T	Treats the designated text as the URL in the first line of an HTTP request header. The supported format is [<protocol>://<hostname>]<path>?<query>, and the text mode is set to URLENCODED by default.

Function	Description
<prefix>.TYPECAST_NUM_T(<type>)	Casts numeric string data to a signed 32-bit number. The argument <type> can be one of the following: • DECIMAL. Treat the string as a decimal number and cast to a signed 32-bit number. • HEX. Treat the string as a hexadecimal number and cast to a signed 32-bit number. • DECIMAL_PREFIX. Consider the part of the string up to the first occurrence of a character that is not a valid decimal character and cast to a signed 32-bit number. • HEX_PREFIX. Consider the part of the string up to the first occurrence of a character that is not a valid hexadecimal character and cast to a signed 32-bit number.

 © Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

1. As traffic through the NetScaler system is evaluated by the different features, each feature performs policy evaluation. For the majority of the features in the packet processing flow, the order of evaluation does not matter because if a policy matches the traffic, the NetScaler system stores the action and continues processing. However, some features need to be first in the packet processing flow because if it matches these, it may not continue evaluation. Which three features should come first in the packet processing flow? (Choose three.)
    a. Application firewall
    b. Caching
    c. Content filtering
    d. Load balancing
    e. Responder
    f. Response rewrite
2. Each pattern in the set has a unique index value. If you do not specify an index for the first pattern, NetScaler assigns the pattern an index of one, which you can change at any time.
    a. True
    b. False

© Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

Indicate whether each statement is true or false.

Statement	True or False
You can bind a classic compression policy either at the global level or at the virtual server level.	
You can bind a given compression policy to more than one bind point.	
The priority values assigned to a policy that is bound to various bind points must have the same value.	
The NetScaler begins with the policy that has the lowest priority value regardless of the bind point to which the policy is bound.	

 © Copyright 2012 Citrix Systems, Inc.

Rewrite, Responder, and URL Transformation

Rewrite

Rewrite refers to the rewriting of some information in the requests or responses handled by the NetScaler system. Rewriting can help in providing access to the requested content without exposing unnecessary details about the web site's actual configuration. A few situations in which the rewrite feature is useful are described below:

- To improve security, the NetScaler can rewrite all the http:// links to https:// in the response body.
- In the SSL offload deployment, the insecure links in the response have to be converted into secure links. Using the rewrite option, you can rewrite all the http:// links to https:// to ensure that the outgoing responses from NetScaler to the client have the secured links.
- If a web site has to show an error page, you can show a custom error page instead of the default 404 Error page.
- If you want to launch a new web site but use the old URL, you can use the rewrite option.
- When a topic in a site has a complicated URL, you can rewrite it with a simple, easy-to-remember URL.
- You can append the default page name to the URL of a web site.

When you enable the rewrite feature, NetScaler can modify the headers and body of HTTP requests and responses.

For more information about the rewrite feature, including rewrite action and policy examples, see Citrix eDocs at *http://edocs.citrix.com*.

Responder

With the responder feature, responses to HTTP requests can be based on who sends the request, where it is sent from, and other criteria with security and system management implications. When users request a web site's home page, you may want to provide a different home page depending on where each user is located, which browser the user is using, or which language(s) the browser accepts and the order of preference. You might want to break the connection immediately if the request is coming from an IP range that has been generating DDoS attacks or initiating hacking attempts.

Responder can be used for the following scenarios depending on client parameters:

- Redirecting an HTTP request to new web sites or web pages
- Responding with some custom response
- Dropping or resetting a connection at request level

For more information about the responder feature, see Citrix eDocs at *http://edocs.citrix.com*.

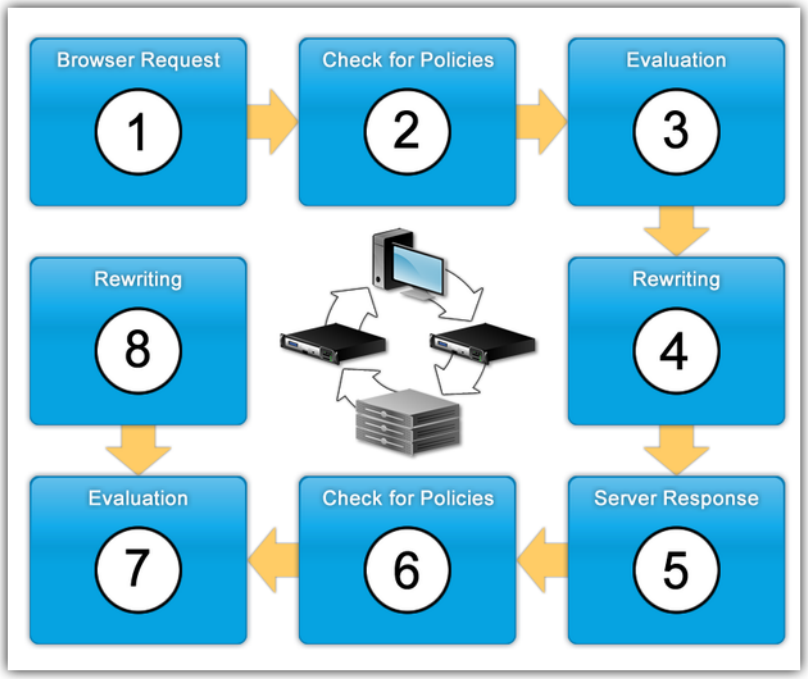
URL Transformation

The URL transformation feature provides a method for modifying all URLs in designated requests from an external version seen by outside users to an internal version seen only by your web servers and IT staff. You can redirect user requests seamlessly without exposing your network structure to users. You can also modify complex internal URLs that users may find difficult to remember into simpler, more easily remembered external URLs.

Before you can use the URL transformation feature, you must enable the rewrite feature.

For more information about URL transformation, see Citrix eDocs at *http://edocs.citrix.com*.

Rewrite Process



The following steps describe the rewrite process:

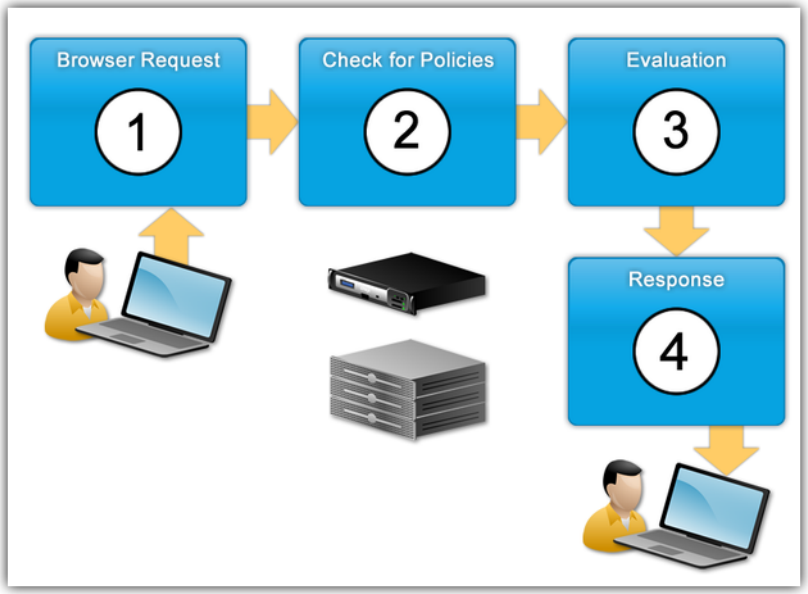
1. The client browser sends a request to the web server through the NetScaler system.

 © Copyright 2012 Citrix Systems, Inc.

2. The NetScaler system checks the request time policy bank for applicable policies.
3. The NetScaler system builds a set of actions to apply after evaluating the list of prioritized policies.
4. The NetScaler system rewrites the request and forwards it to the web server.
5. The web server receives the request and sends a response.
6. The NetScaler system checks the response time policy bank for applicable policies.
7. The NetScaler system builds a set of actions to apply after evaluating the list of prioritized policies.
8. The NetScaler system rewrites the response and forwards it to the client browser.

Once rewrite rules and policies are configured, enabled, and bound to the proper policy bank, the NetScaler system begins to apply those policies. When the browser of an end user sends a request to the web server, the NetScaler system checks the request time policy bank. If the NetScaler system finds rewrite policies, the system evaluates each policy in order of priority, starting with the lowest number and proceeding to the highest number.

Responder Process

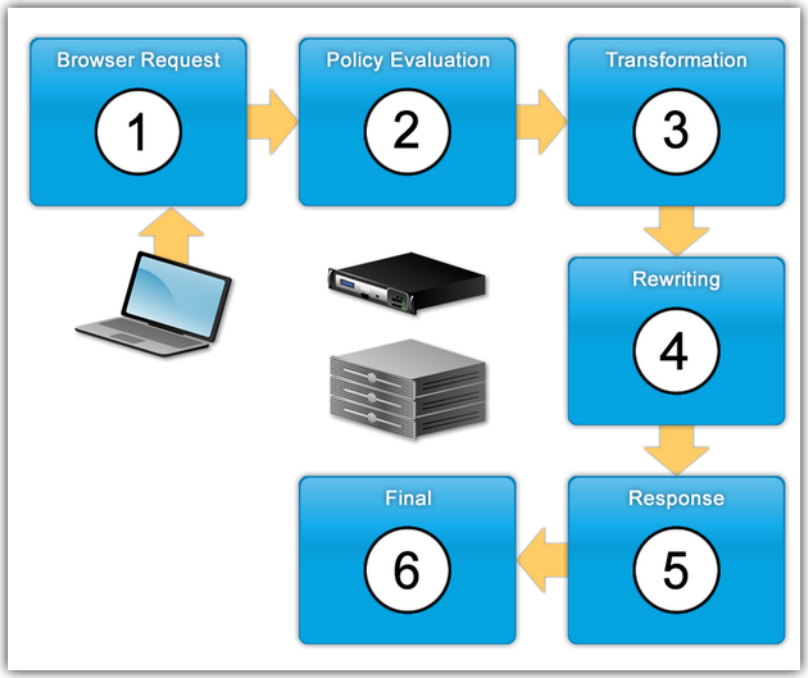


The following steps illustrate the responder process:

1. The client browser sends a request to the web server through the NetScaler system.
2. The NetScaler system checks the request time policy bank for applicable policies.

3. The NetScaler system builds a set of actions to apply after evaluating the list of prioritized policies.
4. The NetScaler system responds to the client request with either a redirect or a respondwith.

URL Transformation Process



The following steps describe the URL transformation process:

1. The client browser sends a request to the web server through the NetScaler system.
2. The policy is evaluated during the request for the entire transaction.
3. The NetScaler creates the transformed/internal URL from the original/external one.
4. The NetScaler rewrites request headers among the last request operations.
5. The NetScaler reverts response headers back to their original state among the first response operations.
6. The NetScaler reverts URLs in the page body.

 © Copyright 2012 Citrix Systems, Inc.

Transformed Elements

Both relative and absolute URLs are transformed. Variables start at 1 for the transformation process and logic.

Elements transformed during the request include:

- Method header (for example GET and POST)
- Host header
- Referrer header

URL transformation does not modify the request body.

Elements transformed during the response include:

- Location header
- Content-location header
- P3P header
- Set-cookie header
- URL attributes under HTML tags in the body, such as href, src, and codebase

Non-HTML content, like JavaScript and CSS, is not transformed.

Configuring Policies and Actions

Rewrite and responder policies are configured using the advanced policy engine. These policies must have an action and a rule and must be bound to a bind point. After the policies are bound to a bind point, the NetScaler system performs rewrite actions.

Configuring Rewrite or Responder Policies

Use the following steps to configure a rewrite or responder action and policy:

1. Define an action to be performed.
2. Create a policy.
3. Define the rule, or expression, which determines when to apply the action.
4. Attach the action for the outcome of the evaluation.
5. Bind the policy (rule + action) to a bind point to perform rewrite.
6. Attach a priority to each policy, which determines the sequence of policy execution.
7. Optional: Refer to the next policy to be evaluated using a goto expression.
8. Optional: Invoke a policy label.

 © Copyright 2012 Citrix Systems, Inc.

Rewrite Default Actions

An undefined event is triggered when the NetScaler cannot evaluate a policy, usually because it detects a logical or other error in the policy or an error condition on the NetScaler. When the rewrite policy evaluation results in an error, the specified undefined action is carried out. Undefined actions configured at the rewrite policy level are carried out before a globally configured undefined action.

The NetScaler supports following three types of undefined actions:

undefAction NOREWRITE	Aborts rewrite processing, but does not alter the packet flow. This means that the NetScaler continues to process requests and responses that do not match any rewrite policy, and eventually forwards them to the requested URL unless another feature intervenes and blocks or redirects the request. This action is appropriate for normal requests to your web servers and is the default setting.
undefAction RESET	Resets the client connection. This means that the NetScaler tells the client that it must re-establish its session with the web server. This action is appropriate for repeat requests for web pages that do not exist, or for connections that might be attempts to hack or probe your protected web sites.
undefAction DROP	Silently drops the request without responding to the client in any way. This means that the NetScaler simply discards the connection without responding to the client. This action is appropriate for requests that appear to be part of a DDoS attack or another sustained attack on your servers.

Undefined events can be triggered for both request and response flow specific policies.

For more information about configuring the default rewrite actions, see Citrix eDocs at *http://edocs.citrix.com*.

Configuring a Rewrite Action

After enabling the rewrite feature, you need to configure one or more actions unless a built-in rewrite action is sufficient. To configure a rewrite action, you assign it a name, specify an action type, and add one or more arguments specifying additional data. The following table describes the more common action types and the arguments you use with them.

Action types that can be used only for HTTP rewrite are identified in the Rewrite Action Type column.

Rewrite Action Type	Argument 1	Argument 2
INSERT_HTTP_HEADER: Inserts the HTTP header you specify into the HTTP request or response. This is the default choice. This action type can be used only with HTTP requests and responses.	The HTTP header you want to insert. For example, if you want to insert the client IP from which a request is sent, type Client-IP.	A string expression that describes the contents of the header you want to insert. For example, if you want to insert the Client IP from which a request is sent, type CLIENT.IP.SRC.
INSERT_BEFORE: Inserts a new string before the designated string.	A string expression that describes the string before which you want to insert a new string. For example, if you want to find the hostname www.example.com and insert a string before the example.com portion, type the following: HTTP.REQ.HOSTNAME.BEFORE_STR ("example.com").	A string expression that describes the new string you want to insert. For example, if you want to insert the new string en. before the string example in the hostname, type en followed by a period.
INSERT_AFTER: Inserts a new string after the designated string.	A string expression that describes the string after which you want to insert a new string. For example, if you want to find the hostname www.example.com and insert a string after the www. portion, type the following: HTTP.REQ.HOSTNAME.AFTER_STR ("www.").	A string expression that describes the new string you want to insert. For example, if you want to insert the new string en. after the string www. in the hostname, type en followed by a period.

 © Copyright 2012 Citrix Systems, Inc.

Rewrite Action Type	Argument 1	Argument 2
REPLACE: Replaces the designated string with a different string.	A string expression that describes the string you want to replace with a new string. For example, if you want to replace the entire hostname in the host header, type HTTP.REQ.HOSTNAME.SERVER.	A string expression that describes the new string you want to insert. For example, if you want to replace the current host header with the string web01.example.net, type web01.example.net.
REPLACE_HTTP_RES: Replace the HTTP response with the value specified in the target field. This action type can be used only with HTTP requests and responses.	A string expression that describes the string with which you want to replace the HTTP response. For example, type HTTP 200 OK You are not authorized to view this page to replace the entire HTTP response with this warning.	
DELETE: Deletes the designated string.	A string expression that describes the string you want to delete. For example, if you want to find and delete the string .en in the hostname of HTTP response headers, type the following: HTTP.RES.HEADER("Host").SUBSTR("en.").	
DELETE_HTTP_HEADER: Deletes the designated HTTP header, including all header contents. This action type can be used only with HTTP requests and responses.	The name of the HTTP header you want to delete. For example, if you want to delete the cache-control header from HTTP responses, type HTTP.RES.HEADER ("Cache-Control").	
DELETE_ALL: Delete every occurrence of the pattern specified in the target text reference.	The part of either the HTTP request or response where you want the deletion to occur.	A string pattern after which the deletion should occur.

Example of NetScaler Command Prompt to Create a New Rewrite Action

At the NetScaler command prompt, type the following commands to create a new rewrite action and verify the configuration:

```
add rewrite action insertact INSERT_HTTP_HEADER "client-
IP" CLIENT.IP.SRC
```

```
show rewrite action insertact
```

Rewrite Action Parameters

The following parameters will be used to configure a rewrite action:

name	A name for your new action or the name of the existing action you want to modify or remove. The name can begin with a letter, number, or the underscore symbol, and can consist of from 1 to 127 letters and numbers and the hyphen (-), period (.), pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should choose a name that will make it easy for others to tell what this action is supposed to do. (This cannot be changed for an existing action.)
pattern or patset	An expression that describes the rewrite operation itself.
bypassSafetyCheck	Whether to bypass the built-in safety checks when adding or modifying this action. Values: YES, NO. Default: NO. You can bypass the safety check for actions that would otherwise generate error messages.
target	A NetScaler default syntax expression that describes the text to be rewritten by the rewrite action. For TCP rewrite actions, the target expression must begin with either CLIENT.TCP.PAYLOAD or SERVER.TCP.PAYLOAD.

 © Copyright 2012 Citrix Systems, Inc.

stringBuilderExpr	Expression specifying the new value of the rewritten packet. Maximum length of a string literal that can be used inside the expression is 255 characters. A string literal that contains more than 255 characters can be split into smaller chunks of 255 characters each. The chunks can then be concatenated with the + operator. Maximum length of the input expression is 8191.
search	Searches for the designated string or expression in the HTTP header or body. You can use a search expression with actions of the following types: delete_all, insert_after_all, insert_before_all, and replace_all.
refineSearch (-refineSearch "extend(#,#).<expression>)	A means of more efficiently searching and rewriting a lengthy response than with a simple search command. Instead of using the search command to search for all occurrences of a regular expression and then rewriting the matched strings, refineSearch enables you to search for a simple string that appears within the text you want to rewrite, refine that search by including surrounding context, and then search only the selected text and context for a regular expression match.

For more information about parameters for configuring a rewrite action, see Citrix eDocs at *http://edocs.citrix.com*.

To Configure a Rewrite Action Using the Configuration Utility

1. In the navigation pane, expand **Rewrite** and then click **Actions**.
2. In the details pane, do one of the following:
   a. To create a new action, click **Add**.
   b. To modify an existing action, select the action and then click **Open**.
3. In the **Add Rewrite Action** or **Configure Rewrite Action** dialog box, specify values for the parameters.
4. Click **Create** or **OK**.
   A message appears in the status bar stating that the action has been configured successfully.
5. Repeat steps 2 through 4 to create or modify as many rewrite actions as necessary.
6. Click **Close**.

Create a Rewrite Action Using the Command-line Interface

At the NetScaler command prompt, type the following commands to clear persistence sessions and verify the configuration:

```
add rewrite action name type target [<stringBuilderExpr>] [(-
pattern
<expression> | -patset string)] [-bypassSafetyCheck (YES|NO)]
```

```
show rewrite action name
```

Example of NetScaler Command Prompt to Create a Rewrite Action

```
add rewrite action insertact INSERT_HTTP_HEADER "client-
IP" CLIENT.IP.SRC
```

```
show rewrite action insertact
```

 © Copyright 2012 Citrix Systems, Inc.

Rewrite Policies

A rewrite policy consists of a rule and an action. The rule determines the traffic on which rewrite is applied and the action determines the action to be taken by the NetScaler. You can define multiple rewrite policies. For each policy, specify the bind point and priority.

A bind point refers to a point in the traffic flow at which the NetScaler examines the traffic to verify whether any rewrite policy can be applied to it. You can bind a policy to a specific load-balancing or content-switching virtual server, or make the policy global if you want the policy to be applied to the entire traffic handled by the NetScaler. These policies are referred to as global policies.

In addition to the user-defined policies, the NetScaler has some default policies. You cannot modify or delete a default policy.

Responder Default Actions

The NetScaler system generates an undefined event (UNDEF event) when a request does not match a responder policy, and then carries out the default action assigned to undefined events. By default, that action is to forward the request to the next feature without changing it. This default behavior is normally what you want; it ensures that requests that do not require special handling by a specific responder action are sent to your web servers and that clients receive access to the content that they requested.

If the web sites your NetScaler system protects receive a significant number of invalid or malicious requests, however, you may want to change the default action to either reset the client connection or drop the request. In this type of configuration, you would write one or more responder policies that would match any legitimate requests, and simply redirect those requests to their original destinations. Your NetScaler system would then block any other requests as specified by the default action you configured.

You can assign any one of the following actions to an undefined event:

NOOP	The NOOP action aborts responder processing but does not alter the packet flow. This means that the system continues to process requests that do not match any responder policy, and eventually forwards them to the requested URL unless another feature intervenes and blocks or redirects the request. This action is appropriate for normal requests to your web servers and is the default setting.
RESET	If the undefined action is set to RESET, the system resets the client connection, informing the client that it must re-establish its session with the web server. This action is appropriate for repeat requests for web pages that do not exist, or for connections that might be attempts to hack or probe your protected web sites.
DROP	If the undefined action is set to DROP, the system silently drops the request without responding to the client in any way. This action is appropriate for requests that appear to be part of a DDoS attack or other sustained attack on your servers.

UNDEF events are triggered only for client requests. No UNDEF events are triggered for responses.

For more information about setting the responder default action, see Citrix eDocs at *http://edocs.citrix.com*.

 © Copyright 2012 Citrix Systems, Inc.

Responder Actions

After enabling the responder feature, you must configure one or more actions for handling requests.

The responder supports the following types of actions:

Respond with	Sends the response defined by the Target expression without forwarding the request to a web server. (The NetScaler system substitutes for and acts as a web server.) Use this type of action to manually define a simple HTML-based response. Normally the text for a respond with action consists of a web server error code and a brief HTML page.
Respond with SQL OK	Sends the designated SQL OK response defined by the Target expression. Use this type of action to send an SQL OK response to an SQL query.
Respond with SQL Error	Sends the designated SQL Error response defined by the Target expression. Use this type of action to send an SQL Error response to an SQL query.
Respond with HTML page	Sends the designated HTML page as the response. You can choose from a drop-down list of HTML pages that were previously uploaded, or upload a new HTML page. Use this type of action to send an imported HTML page as the response.
Redirect	Redirects the request to a different web page or web server. A redirect action can redirect requests originally sent to a “dummy” web site that exists in DNS, but for which there is no actual web server, to an actual web site. It can also redirect search requests to an appropriate URL. Normally, the redirection target for a redirect action consists of a complete URL.

For more information about configuring a responder action, see Citrix eDocs at *http://edocs.citrix.com*.

Respondwith

A respondwith action is used to send an HTTP string as a response to a client.

The following table describes the parameters you must configure in order to create a respondwith action.

Parameter	Description
Name	The name of the action. This is a mandatory parameter and cannot be changed. The action name must not exceed 31 characters.
Type	The type of responder action being configured. This parameter determines the behavior of the responder action. You can choose to respond to the client or redirect the client elsewhere. Select the 'respondwith' responder action type to send the configured target string to the client.
Target	This parameter contains the HTTP string to be sent as a response to the client.

To Create a Respondwith Action

1. In the left pane, expand **Protection Features**, then expand **Responder** and click **Actions**.
   The Responder Actions page appears in the right pane.
2. Click **Add**.
   The Create Responder Action dialog box appears.
3. In the Name text box, type the name of the responder action.
4. Under Type, select **Respond with**.
5. In the Target text area, type `"HTTP/1.1 200 OK\r\n\r\n" + "Client: " + CLIENT.IP.SRC + " is not authorized to access URL:" + HTTP.REQ.URL.HTTP_URL_SAFE`.
6. Click **Create**, then click **Close**.
   The responder action you created appears in the Responder Actions page.

 © Copyright 2012 Citrix Systems, Inc.

Responder Action for Timeouts

You can invoke a responder action when an HTTP request times out. To configure this feature, first create the responder action that you want to invoke. Then, configure the global HTTP timeout action.

To configure the global HTTP timeout action to invoke a responder action by using the NetScaler command line, type the following command:

```
set ns httpProfile -reqTimeoutAction responder action name
```

Responder Policies

Responder policies are configured in the configuration utility and in the command-line interface. The following arguments are identified when adding a responder policy:

Rule	Expression used by the responder policy that returns a Boolean result
Action	Responder action name, NOOP or RESET
UndefAction	Responder action to be taken in the case of an undefined event during policy evaluation, with value of NOREWRITE or RESET

 © Copyright 2012 Citrix Systems, Inc.

Responder HTML Page Imports

The responder feature can respond to designated requests by sending the client an HTML-based web page that you upload to the NetScaler system. You have the option of redirecting the request or responding with a response code and answer configured on the NetScaler itself.

To use this feature, first upload an HTML-based web page to the NetScaler by using either the NetScaler command line or the configuration utility. Next, configure a responder action with type set to RespondWithHTMLPage and the name of the HTML page. Finally, create a responder policy and bind it to the action.

To upload an HTML page to the responder feature, at the NetScaler command prompt type the following commands:

```
import responder htmlpage [src] name [-comment comment] [-
overwrite]
```

For more information about configuring a responder action, see Citrix eDocs at *http://edocs.citrix.com*.

Binding Policies

Only bound policies respond or rewrite on the NetScaler system. Binding occurs in the configuration utility and in the command-line interface.

Each policy needs a priority assigned to it that is a positive integer constant. A lower value priority number is interpreted as a higher priority. For example, a priority value set to 10 is a higher priority than a priority value set to 30. Duplicate priorities are not allowed within each bind point.

Possible GotoPriorityExpression values include:

END	Terminates policy evaluation and proceeds to apply the action
NEXT	Proceeds to the next policy in the priority ranking
Positive integer	Proceeds to the policy with the priority ranking as the specified type, indicating the type of global bind point

 © Copyright 2012 Citrix Systems, Inc.

Configuring URL Transformation

The URL transformation feature identifies URL patterns in HTML pages and modifies them to a different form by translating URLs from their external, public appearance to an internal, hidden resource. This feature simplifies the external appearance of web sites, hides back-end applications and architecture, and allows infrastructure changes without impacting the user experience. It also redirects links to third-party service providers and reduces the number of broken links.

To plan for URL transformation, identify the transformed elements:

- URL transformation policy: When to perform transformation
- URL transformation profile: Where to find URLs and actions
- URL transformation action: How to transform and what to do

URL Transformation Actions

The URL transformation action fields and the variables that can be used include:

- Configuring URL transformation actions
- Troubleshooting URL transformation

Configuring URL Transformation Actions

The request and response from fields:

- Contain PCRE expressions to match URLs
- Up to five back-references (wildcards)

When configuring the request into and response into field, you can:

- Use the text field with $ (dollar sign) as a special character to access wildcard data by its number
- Use $$ for a literal dollar character

Cookie domain into field is a text field. You can use $ (dollar sign) as a special character to access wildcard data in the Response URL From field.

It is possible to transform only the request or the response side of a transaction.

URL Transformation Procedure

To configure URL transformation:

1. Verify the rewrite feature is enabled.
2. Create a URL transformation profile.
3. Add actions to the profile.
4. Create a URL transformation policy.
5. Bind the policy.

Troubleshooting URL Transformation

You can enable the debug logs for URL transformation to assist in troubleshooting issues with the feature configuration.

Use the following tools to troubleshoot URL transformation:

- Debug log messages

 © Copyright 2012 Citrix Systems, Inc.

- Memory statistics for URL transformation using nsconmsg

Binding of URL Transformations

After you have configured your URL transformation policies, you globally bind them to put them into effect. After global binding, any request or response that matches a URL transformation policy is transformed by the profile associated with that policy.

When you bind a policy, you assign a priority to it. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer. In the NetScaler OS, policy priorities work in reverse order, so the higher the number, the lower the priority.

Setting policy priorities is important for achieving the intended results. The URL transformation feature implements only the first policy that a request matches. It will not implement any additional policies that it might also match. For example, if you give your first policy a low priority (such as 1000), you are instructing the NetScaler to perform it only if other policies with a higher priority do not match a request. If you give your first policy a high priority (such as 1), you are instructing the NetScaler to perform it first and skip any other policies that might also match. If you set priorities with intervals of 50 or 100 between each policy when you globally bind your policies, you can easily add other policies in any order without reassigning priorities.

For more information about globally binding URL transformation policies, see Citrix eDocs at *http://edocs.citrix.com*.

 © Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

1. Which feature would be used to modify a complex internal URL into a simpler external URL? (Choose one.)
    a. Rewrite
    b. Respond
    c. URL transformation
2. An undefined event can be triggered for _____. (Choose the correct answer.)
    a. request policies
    b. response policies
    c. URL transformation policies
    d. request and response policies
    e. request, response, and URL transformation policies
3. Which three responder actions can you assign to an undefined event? (Choose three.)
    a. DELETE
    b. DELETE ALL
    c. DROP
    d. INSERT
    e. NOOP
    f. RESET

© Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

Match the following terms with the correct descriptions:

- Rewrite
- Responder
- URL transformation

Description	Term
When users request a web site's home page, this will allow you to provide a different home page depending on where the user is located, which browser the user is using, or which language the browser accepts.	
This identifies URL patterns in the HTML pages and modifies them to a different form by translating URLs from their external, public appearance to an internal, hidden resource.	
This can help in providing access to the requested content without exposing unnecessary details about the web site's actual configuration.	

 © Copyright 2012 Citrix Systems, Inc.

Module 12

Optimizing Traffic

© Copyright 2012 Citrix Systems, Inc.

Overview

The NetScaler optimization features reduce transaction times between the clients and the servers and reduce bandwidth consumption. The optimization features also enhance server performance by offloading some tasks and making others more efficient.

After completing this module, you will be able to:

- Implement compression policies to optimize traffic.
- Explain and configure the integrated caching feature.
- Cache frequently served web content to relieve servers and reduce response time.
- Use templates to optimize web traffic from specific applications.
- Route traffic based on specific parameters.

© Copyright 2012 Citrix Systems, Inc.

Introduction to Compression

By default, the NetScaler system compresses text/HTML and text/plain Multipurpose Internet Mail Extensions (MIME) formats for all browsers. For other (text/*) MIME formats, the NetScaler system compresses traffic based on the format supported by the browser.

The NetScaler system can compress content generated by most Common Gateway Interface (CGI) applications. It does not, however, automatically compress images in GIF and JPEG formats as they are already compressed and would not benefit from further compression. Compressing content that is already compressed typically leads to larger files and decreased performance. You can configure the NetScaler system to compress responses based on file size. This threshold is applicable to responses of any content length, but you can set it only if the response consists of one packet.

Compression Process

A NetScaler system configured with compression policies intercepts requests from compression-aware clients and applies the compression policies to determine whether the client can accept compressible content. After the NetScaler system receives the HTTP response from the server, the content is examined again to determine if it is compressible. If the content is compressible, it is compressed and the compressed content is forwarded to the client. The response header is modified to indicate the type of compression performed.

Compression Considerations

The following list identifies considerations for using compression on the NetScaler system.

Compression algorithm	The NetScaler system supports compression through the GZIP/deflate mechanism. This mechanism is the default for the NetScaler system and is an open standard described in RFCs 1950, 1951, and 1952.
Compression ratio	HTML content is compressed to a typical ratio of 3:1 with a data size of 32 KB available for compression. The exact ratio depends on the type of data and the global compression setting. Data with a significant amount of formatting and URL references compresses more effectively. Similarly, a 20 KB page yields a better compression ratio than a 200 byte page. The increase in compression ratio is not linear with data size.

 © Copyright 2012 Citrix Systems, Inc.

Browser awareness	Only compression-aware browsers are provided with compressed data. Other browsers receive uncompressed data. Configuration is not needed for browser types.
HTTPS compression	HTTPS compression is supported when encryption is offloaded by the NetScaler system. The responses of the server are compressed and encrypted before they are sent to the HTTPS client.
HTTP versions	Both HTTP versions 1.0 and 1.1 are supported with GZIP/deflate in the accept-encoding header.

Compression Responses

Responses are compressed; requests are not compressed.

> Compression is done only on responses with a 200 status code. Responses for which the content length is zero are not compressed. Additionally, if the response is already compressed, it is detected and bypassed by the compression engine, enabling the originating sites to use existing server-based compression in conjunction with the compression feature of the NetScaler system.

The NetScaler system generates a content-length response only if the full response fits within the NetScaler system internal buffer (size of 56,000 bytes).

Compression Parameters

Compression parameters include:

Compression level	Specifies the level of compression to be carried out. Options for compression include: optimal, best speed, and best compression.
Quantum size	Specifies the minimum quantum of data to be filled, after which data begins to compress. The default quantum size is 57,344 bytes. The quantum size can vary between 1 and 63,488.

Threshold ratio for heuristic expiry	Specifies the threshold compression ratio for heuristic base file expiration, multiplied by one-hundredth. For example, 125 indicates a threshold ratio of 1.25. The default threshold ratio is 100, and the range is from 1 to 1,000.
History weightage for heuristic expiry	Specifies the weightage-to-delta-compression history for the heuristic base file expiration as a percentage. The default value is 50, and the range is 1 to 100.
Minimum HTTP-response size	Specifies the minimum file size to be compressed. The default value is 0, meaning that all HTTP responses are compressed. The value is expressed in bytes. This parameter helps eliminate CPU usage for the small files that do not benefit from compression.
Bypass compression on CPU usage	Specifies the percentage of CPU level at which the NetScaler system bypasses the compression feature while processing traffic.
Server-side compression	Enables compression on the server. When this is enabled, the data transfer between the NetScaler system and the server is compressed. This parameter is enabled by default.
Heuristic base file expiry	Enables heuristic base file expiration.

Compression Policies

Compression policies enable the NetScaler system to identify the content that needs to be compressed. Each compression policy consists of an expression and an action. An expression is created to identify the files entering the system (for example, HTML files, ASP files, or PDF files). An action defines the action the system performs on the file identified by the expression. For example, you can configure a compression policy consisting of an expression that identifies the PDF files and an action that compresses the PDF files.

A deployment in which compression policies are bound globally enables the compression policies to act on any service regardless of the virtual server to which the service is bound. Compression policies can also be bound to a load-balancing virtual server. The compression policies are then evaluated for only the services bound to the virtual server. The virtual server identifies the least-loaded originating server and directs client requests to the corresponding service.

 © Copyright 2012 Citrix Systems, Inc.

Disabling the compression feature will disable compression on the entire NetScaler system even if it is enabled at the service level.

Setting Compression Policies Globally

Either the classic or advanced compression global policy set can be used for compression policy evaluation for entities:

- HTTP/HTTPS load-balancing or content-switching virtual servers with no compression policies bound
- Transparent HTTP/HTTPS services

Control the selection by using the NetScaler command-line interface and enter:

```
set cmp parameter –policyType (CLASSIC | ADVANCED)
```

Virtual servers that have no compression policies bound use the globally bound classic compression policies by default.

Classic compression polices will not be converted to their advanced equivalents during an upgrade; they should be manually converted after the upgrade, if required.

Configuring Compression

Configure compression in the configuration utility or in the command-line interface. Policy labels are used to invoke the compression policies. The following demonstrates the commands and syntax used to manage advanced compression policy labels:

```
add cmp policylabel labelName -type ( REQ | RES )
```

```
rm cmp policylabel labelName
```

```
bind cmp policylabel labelName -policyName string -
priority positive_integer
-gotoPriorityExpression expression -invoke labelType labelName
```

```
unbind cmp policylabel labelName policyName -
priority positive_integer
```

```
show cmp policylabel labelName
```

You can use the same commands to manage classic compression policies and advanced compression policies.

Compression Actions

You associate actions with compression policies. If an HTTP request matches the policy rule, the action is applied to the response. There are four built-in compression actions:

COMPRESS	Uses the GZIP algorithm to compress data for browsers that support either GZIP or both GZIP and DEFLATE. If the browser does not support either algorithm, and the action has been set to COMPRESS, the NetScaler system does not compress data.
NOCOMPRESS	Disables the compression of data.
GZIP	Uses the GZIP algorithm to compress data for browsers that support GZIP compression. If GZIP compression is not supported on the browser, the system does not compress data.
DEFLATE	Uses the DEFLATE algorithm to compress data for browsers that support DEFLATE compression. If DEFLATE compression is not supported on the browser, then the system does not compress the data.

 © Copyright 2012 Citrix Systems, Inc.

Integrated Caching

Integrated caching stores frequently requested content in the memory cache of the NetScaler system. It intercepts all HTTP client requests and sends the response to the client if the response is stored in the memory cache. Responses for requests may or may not be stored in the cache.

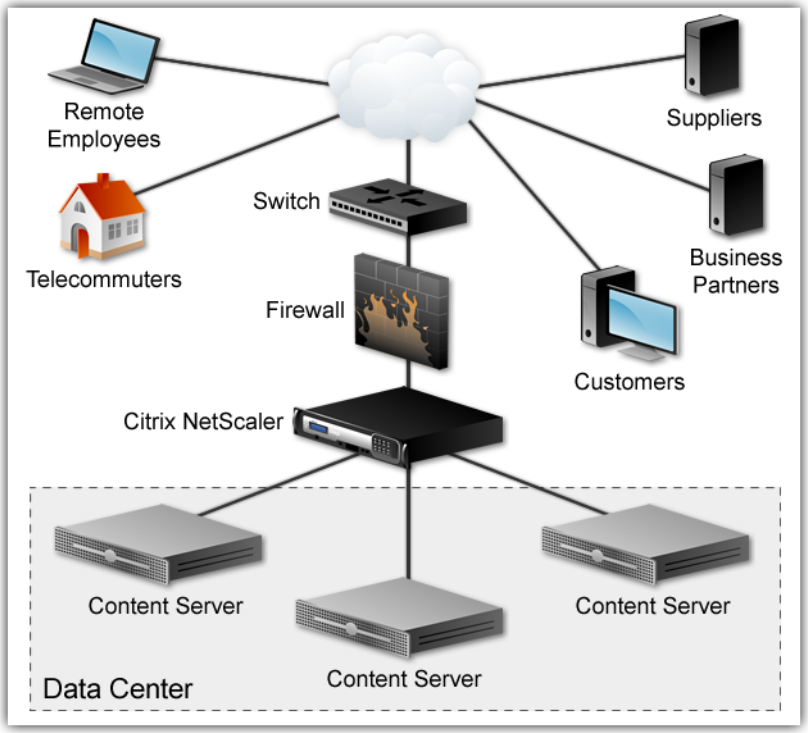
NetScaler integrated caching offers:

- HTTP caching: HTTP and (SSL-offloaded) HTTPS traffic only
- HTTP/1.1 and HTTP/1.0 only
- Fine-grained configuration supporting static and dynamic content
- Seamless integration with other NetScaler system features
- Fast in-memory cache
- Usage of up to 50 percent of the physical memory on the system
- Easy manageability

Reverse-Proxy-Cache Configuration

As the figure illustrates, a reverse proxy resides in front of one or more web servers and shields the origin server from client requests. Often, a reverse proxy cache is a front end for all client requests to a server. All connections coming from the Internet addressed to one of the content servers are routed through the proxy server, which may deal with either the request itself or pass the request wholly or partially to the main content server. It is useful in an integrated-caching solution to have the NetScaler system serve as a reverse proxy.

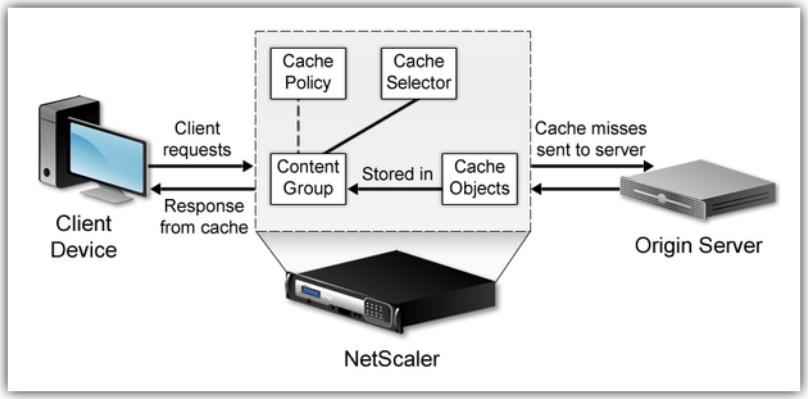
In the reverse-proxy mode, when the NetScaler system receives a request, a cache-redirection virtual server evaluates the request and forwards it to either a load-balancing virtual server for the cache or a load-balancing virtual server for the origin.



 © Copyright 2012 Citrix Systems, Inc.

Content Groups

A content group is a container for cached objects that can be served in a response. Every cached object is a member of a content group, as illustrated in the figure. Content groups can be used to control common settings for content cached within that group, such as content-expiry method, prefetch, cookie removal, and header modification settings.



Cache Selectors and Policies

Cache selectors

Cache selectors are patterns or expressions that are formed from a combination of different parts of a request. When a client requests dynamic content, selectors search through the cache to locate the appropriate response for the client.

Policies

Policies are the combination of an action and an expression to determine which requests and responses should be cached and in which cache-content group the object will be stored. Policies can also be used to invalidate cached objects or content groups.

 © Copyright 2012 Citrix Systems, Inc.

Caching Static and Dynamic Content

Integrated caching is capable of caching dynamically generated content, which is typically marked non-cacheable by web servers and application servers. Static content remains the same for multiple users, including page-based caching such as search engine pages, and object-based caching, such as web-based application graphics. Dynamic content periodically changes, which includes object-based caching only, such as stock updates, sports scores, and news.

Dynamic Content

Caching dynamic web application content requires complex configuration. To cache dynamic responses, you must understand the application behavior and determine which responses are cacheable. Most applications declare almost every dynamic response as non-cacheable, but any response that does not change with every request is potentially cacheable. Dynamic content requires a time-based refresh to maximize content caching.

You can select one of the following parameters to cache dynamic objects:

- The duration for which the response can be cached
- The trigger that causes the response to change

If configuring a trigger that causes a response or change, then you should determine whether that trigger is used to flush the object from the cache. Dynamic content caching often requires parameterized caching.

Dynamic Content Example

Static Content Example

Cache Actions

The following table lists the cache actions.

Actions	Description
CACHE	This action makes the transaction cacheable. If it is associated with a content group, then the response is stored in that content group. Otherwise, it is stored in the DEFAULT content group.

© Copyright 2012 Citrix Systems, Inc.

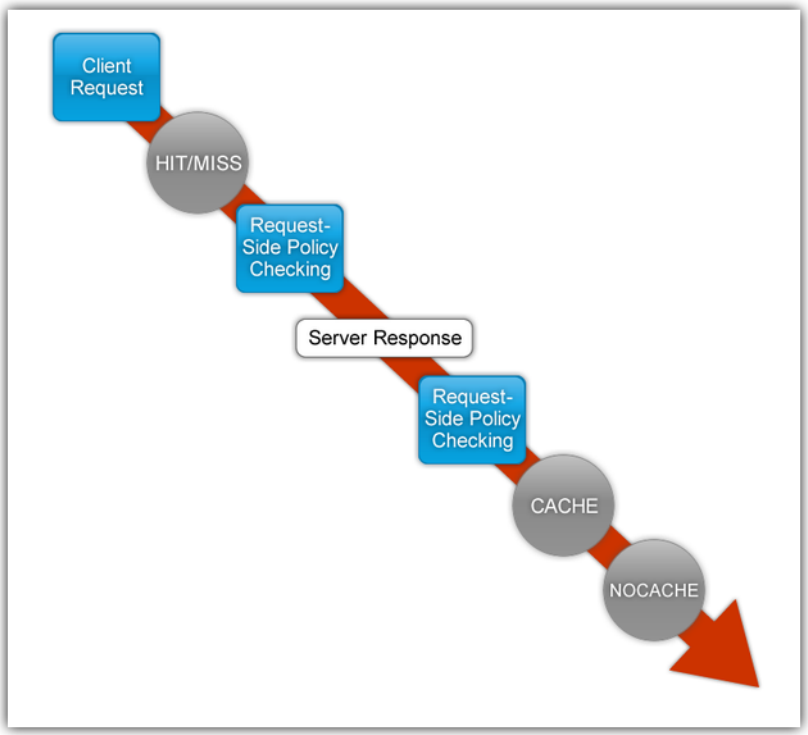
Actions	Description
NO CACHE	This action makes the transaction non-cacheable. The response is not stored in the cache.
MAY_NOCACHE	If neither request- nor response-cache policies match, then the default action is to cache the response. The MAY_NOCACHE action reverses this behavior for chosen requests. This action, associated with a request-cache policy, makes the response non-cacheable if no response cache policy is matched. The MAY_NOCACHE request time directive can be overridden by a CACHE response time directive.
MAY_CACHE	This action, associated with a request-cacheability policy, makes the response cacheable if no response-cacheability policy matches, which is very similar to the default behavior. This action can also take the name of a content group. The DEFAULT content group is assumed if no content group name is specified. The response gets cached, if at all, in the specified content group. The MAY_CACHE request-time directive can be overridden by a NOCACHE directive at response time. A CACHE response-time directive without any content group specified results in the response being cached in the content group specified with the MAY_CACHE action. A CACHE response-time directive with a content group different from the content group specified with the MAY_CACHE action results in the response not getting cached.
INVAL	This action invalidates some cached objects. The INVAL action also has the side effect of making a transaction non-cacheable. The assumption is that an invalidation request modifies an object or transaction at the origin, and it cannot be cached.

 © Copyright 2012 Citrix Systems, Inc.

Request and Response Process Flow

The integrated-caching process flow and request, and response-policy-evaluation flows are important in understanding built-in cacheability policies. Two sets of built-in cacheability policies exist. One policy is for request-time processing and the other is for response-time processing. These built-in policies encode the standard HTTP behavior, which is part of the integrated-cache default behavior.

The figure shows how integrated caching occurs early on in the request and response process flow. The client request hits or misses, as well as undergoes request-side policy checking. The server response undergoes response-side policy checking and goes through a CACHE or NOCACHE action.



Request Side

The incoming client request is tested for invalidation policies. If there is a match, the content is not part of a valid content group and is not cached.

If the request does not match a policy, the NetScaler system verifies if the response is in the cache. If it is, then the request is a hit and is sent to the client. If it is not in the cache, the NetScaler

system tests request-side policies to perform cache, no cache, may not cache, or may cache according to the configuration.

 © Copyright 2012 Citrix Systems, Inc.

Cache Policies and Cache Expressions

You can use cache policies and cache expressions to configure integrated caching.

Cache Policies

Cache policies use the powerful NetScaler policy framework. Policies can be used to override and add to the built-in standard HTTP caching behavior. Policies determine what transactions are cacheable. They decide into which content group a cacheable object goes, and they determine which set of objects in the cache get invalidated by a given HTTP request.

In order to configure the NetScaler system for dynamically generated content or a non-standard static object, you need to configure cache policies to encode the desired behavior.

Cache Expressions

The expression specified with the policy is evaluated on the incoming request or response. The expressions are standard classic-policy-engine expressions with one exception. Whereas the syntax, semantics, and operators are the same, an expression can contain request-only expressions or response-only expressions, but not both.

Depending on whether the rules have request-only expressions or response-only expressions, cache policies can be classified as request policies or response policies. Other than controlling when a policy is evaluated, this classification also controls what kind of action can be associated with a policy.

Cache Policies

Cache policies:

- Are either request-side or response-side
- Equal a name + rule + action
- Name a content group in which cached objects are stored
- Are built-in and non-modifiable or are custom and modifiable
- Are HTTP/1.0 and HTTP/1.1 request-cacheable only
- Are only GET-method cacheable
- Are non-cacheable with a response pragma header
- Are non-cacheable with a response set-cookie header

© Copyright 2012 Citrix Systems, Inc.

Binding Cache Policies

Cache policies must be bound to be active. The policies can be bound by using the configuration utility or by using the `bind cache global` command in the command-line interface.

The options for binding cache policies are described below.

Priority

The priority determines the evaluation order of policies. The lower the value specified for the priority option, the higher the priority of the associated policy.

Precede Default Rules

The Precede Default Rules option designates that the policy will be evaluated prior to (and therefore take precedence over) the built-in policies.

> Policies with an INVAL action must be bound with a higher priority than other policies with the Precede Default Rules option selected.

Cache Policy Evaluation

Policies are made up of expressions for policy evaluation and a corresponding action to be applied when the expression is matched. Moreover, policies must be bound with priorities to define their evaluation order when multiple policies are present.

If	Then
A request matches an invalidation policy	That request bypasses the cache.

 © Copyright 2012 Citrix Systems, Inc.

If	Then
Bypass is enabled	The hit evaluation happens only after the pre-default, default, and post-default request-cacheability policies have been evaluated. If bypass is enabled, then the request-time cache policies have already been evaluated to test whether the request should bypass the cache or not. If the matching object is present but has expired, then the cache policies are not evaluated again unless the -alwaysEvalPolicies argument is set to YES in the content group of the expired object.
Bypass is not enabled	Hit evaluation will happen before the request-cacheability policies are evaluated. Not enabling bypass leads to better performance because most hits are served without any cacheability-policy evaluation.
No policies match (built-in or user-configured)	The response is cached in the DEFAULT content group.
The cacheability decision has already been made at request time	No further cacheability-policy evaluation takes place.
A request matches a request-time cacheability policy with a MAY_CACHE or a MAY_NOCACHE action	The request-time-cacheability policies with lower priorities are not evaluated. The response-cacheability policies are still evaluated.

The first CACHE or NOCACHE policy that matches determines whether the response object and the group to which the object belongs are cacheable. Cacheability evaluation happens only for requests that do not find a matching object in the cache, such as cache misses.

Action Analytics

The performance of your web site or application depends on how well you optimize the delivery of the most frequently requested content. Techniques such as caching and compression help accelerate the delivery of services to clients, but you need to be able to identify the resources that are requested most frequently, and then cache or compress those resources. You can identify the most frequently used resources by aggregating real-time statistics about web site or application traffic. Statistics, such as how frequently a resource is accessed relative to other resources and how much bandwidth is consumed by those resources, help you determine whether those resources need to be cached or compressed to improve server performance and network utilization.

The Stream Analytics feature operates on a single NetScaler system and collects run-time statistics on the basis of criteria that you define. When used with NetScaler policies, the feature also provides the infrastructure that you need for automatic, real-time traffic optimization.

When configuring Stream Analytics, you must specify the request attributes for which you want to collect statistical data, for example, URLs and HTTP methods, by configuring default syntax expressions in an entity called a selector. Then, you create an identifier to configure settings such as the sampling interval and sample count. You also configure a policy that enables the system to evaluate traffic as specified by the selector-identifier pair. Finally, you bind the policy to a bind point to begin collecting statistics. The system also provides you with a set of built-in selectors, identifiers, and responder policies that you can use to get started with the feature.

The system aggregates the following statistics:

- Number of requests
- Bandwidth consumed by the requests
- Response time
- Number of concurrent connections

You can configure the feature to perform run-time sorting of the records on an attribute of your choice. You can view the statistical data by using either the command-line interface or the Stream Sessions tool in the configuration utility.

 © Copyright 2012 Citrix Systems, Inc.

Evaluation Order

The following information applies to the evaluation order of cache policies:

- The cache policies are listed in the order in which they are evaluated during the lifetime of an HTTP transaction.
- The response-cacheability policies can have only CACHE or NOCACHE actions. The CACHE action can further specify a non-parameterized content group in which to store the response (otherwise, DEFAULT is assumed).
- The policy-evaluation order is controlled by the priority to which it is bound within each classification. The NetScaler system ensures that you cannot specify inconsistent priority. For example, you cannot specify a priority for a cacheability policy that is higher than the priority of any invalidation policy.
- Two policies cannot have the same priority value.
- The action that becomes effective is the one that is associated with the first policy in the priority order that matches within each classification.

© Copyright 2012 Citrix Systems, Inc.

Built-in Policies

The integrated cache has built-in-request and response-default policies. The NetScaler system evaluates the policies in a particular order. These policies are hard-coded for efficiency. The built-in-policy names start with an underscore (_). You can view the built-in policies from the configuration utility and the command-line interface using the `show cache policy` command.

You can override these built-in policies with a user-defined policy that is bound to a request-time-override or response-time-override policy bank.

 © Copyright 2012 Citrix Systems, Inc.

Graceful Cache Configuration Changes

Integrated caching is on unless disabled, and reconfiguring while integrated caching is enabled can lead to unexpected behavior. To avoid unexpected behavior, you can configure changes gracefully.

To perform graceful changes to the integrated cache:

1. Disable integrated caching.
2. Flush the cache.
3. Make configuration changes.
4. Enable integrated caching.

By default, the caching feature caches all cacheable content when the feature is enabled. You need to configure the caching feature if the default caching behavior is not desired.

Test Your Knowledge

1. When configured for integrated caching, it is useful to have the NetScaler system serve as a _____. (Choose the correct answer.)
   a. content group
   b. host
   c. switch
   d. reverse proxy
2. Which two steps must you complete before reconfiguring integrated caching on the NetScaler system? (Choose two.)
   a. Disable integrated caching
   b. Edit the content groups
   c. Flush the cache
   d. Restart integrated caching

 © Copyright 2012 Citrix Systems, Inc.

Cache Content Groups

Every cached object is made a member of a content group. The association happens at the time the object is being downloaded and stored. This association is declared in the policy that resulted in the caching of this object.

Cache content groups are used to organize content, so that settings may be managed at a group level rather than for each policy.

The following content group names are reserved:

- DEFAULT
- ALL

Reversed Content Group Names

DEFAULT refers to a built-in content group. If a group name is not specified in a cache policy, then all objects cached by that policy are cached in the DEFAULT content group. Although the DEFAULT content group cannot be removed, the DEFAULT group is no different from any other explicitly configured group.

Unless it is changed, the DEFAULT content group uses the standard HTTP mechanism for expiring the objects. The DEFAULT content group is not parameterized and has prefetch enabled.

Changing the settings of the DEFAULT content group does not affect any other group in any way.

The name ALL is also reserved. ALL refers to all content groups, including the DEFAULT group.

Cache Content Aging

Controlling content aging involves calculations regarding the expiry time of an object and is set as an attribute of the cache content group. Expiration types include:

User-configured recurring absolute expiration	The absolute expiry time specifies the exact time when the object must expire every day. The time is specified in the HH:MM format, and either in local time or GMT time. The absolute expiry can be configured to occur up to four times daily.
User-configured recurring relative expiration	The relative expiry time specifies the number of seconds or milliseconds from the time the request missed the origin to the expiry time of the object.

Heuristic expiration	The heuristic expiry time determines the time of expiry by considering the last modified header. The time of expiry is expressed as a percentage of the duration since the object was last modified. The heuristic that remains unmodified for a longer time is likely to have a longer expiry period.
Weak positive relative expiration	The weak positive relative expiry is set in conjunction with heuristic expiration. The setting is used to set relative expiration for responses that do not contain a last-modified header.
Weak negative relative expiration	The weak negative relative expiry is used to set relative expiration for negative responses (such as 404 Page Not Found or 301 Content Moved).

Content Group Settings

Content group settings affect content aging, memory usage, HTTP headers, cookie removal, and other attributes.

Memory Limit

Content group settings that affect memory include:

- Memory limit
- Minimum response size
- Maximum response size
- Pinned

A memory limit can be configured for each content group. This configuration allows you to limit the amount of available cache space that a specific content group can consume.

There is a minimum and maximum response size to cache in the cache content group.

Selecting the pinned option prevents objects in the content group from being flushed under memory pressure.

Objects in the content group that are pinned will not participate in normal object removal procedures.

 © Copyright 2012 Citrix Systems, Inc.

HTTP Headers and Content Group Settings

Each of the following content group settings affects HTTP headers:

Insert Age — If this action is configured, the Age header is inserted into responses served from cache. The Age response-header field conveys the sender's estimate of the amount of time since the response message (or its re-validation) was generated at the origin server.

Insert Via — If this action is configured, the Via header is inserted into responses served from cache. The Via header denotes that the object was served out of the NetScaler cache.

Insert Etag — If this action is configured, the Etag header is inserted into responses served from cache. An Etag is an HTTP response header returned by an HTTP/1.1 compliant web server used to determine change in content at a given URL. When a new HTTP response contains the same Etag as an older HTTP response, the content is determined to be the same without further downloading.

Cache-Control — A Cache-Control header can be specified in the response. Any valid entry for a Cache-Control header can be specified.

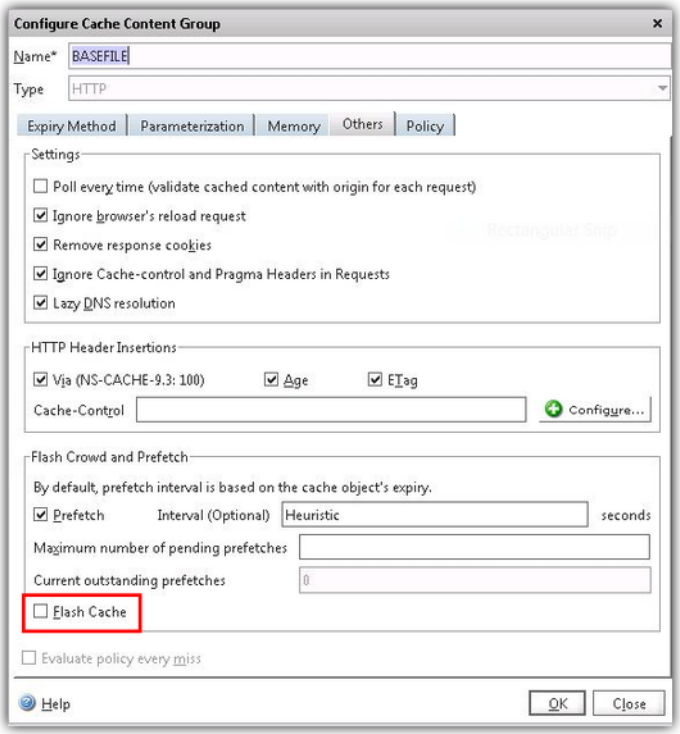
Other Content Group Attributes

The following table describes other content group attributes.

Attributes	Description
Poll every time	This setting forces the system to re-validate all objects with the server before serving a response from the cache.
Prefetch	This parameter is used to decide whether the integrated cache should attempt to refresh an object immediately before it is expired.
Minimum hits	This setting specifies the minimum number of accesses for an object to be stored in cache. The default value is 0.

© Copyright 2012 Citrix Systems, Inc.

Flash Cache



The Flash Cache option queues requests that arrive simultaneously (a flash crowd), retrieves the response, and distributes it to all the clients whose requests are in the queue. When enabled, this option causes the NetScaler system to allow only one request to reach the server. All the other requests are queued until the server responds. The response from the server is then returned to all queued requests, improving flash-crowd handling.

If the originating server responds with an error message, the same error message is repeated across all the recipients. Upon receiving a request, the NetScaler system queues the clients with the assumption that the response is cacheable. If the response becomes non-cacheable while being stored, the NetScaler system aborts storing and continues to send the response from the server to all the queued clients.

The Flash Cache feature can be enabled on a per-content group basis. Poll every time (PET) and Flash Cache cannot be enabled on a single content group. Flash Cache can be used in conjunction with the Expire at Last Byte option if the content should not be retained in cache.

 © Copyright 2012 Citrix Systems, Inc.

Global Cache Attributes

Cache global settings apply to all objects stored in the cache regardless of the content group in which they belong. Cache global settings include:

- ON | OFF
- Via
- memLimit (up to 50% of memory)

Cache Arguments

The `set cache parameter` command modifies the global configuration of integrated cache. You can use the following command to cache parameters in the command-line interface:

```
set cache parameter -memLimit MB -via string -
verifyUsing verifyusing -maxPostLen PositiveInteger -
prefetchMaxPending PositiveInteger -enableBypass ( YES | NO)
```

memLimit

The memory limit for integrated cache is configured by memLimit. The default is no value. Value settings can be set to a minimum of 256 MB and a maximum of 7192 MB.

via

A via header is inserted in all responses served from a content group if its insertVia flag is set.

verifyUsing

The criteria for deciding whether a cached object can be served for an incoming HTTP request is configured with verifyingUsing.

maxPostLen

The maximum number of POST body bytes to consider when evaluating parameters for a content group for which you have configured hit and invalidation parameters. Minimum value is 0, maximum value is 131,072.

prefetchMaxPending

The maximum number of outstanding prefetches in integrated caching is configured with prefetchMaxPending. The default is no value and the maximum value is 4,294,967,294.

enableBypass

The bypass argument is configured with enableBypass. If set to NO, then an incoming request serves a hit if a matching object can be found in the cache storage, regardless of the cacheability policy configuration. If set to YES, then the bound request cacheability policies are evaluated before any hit selection in the cache storage is attempted. If the request matches a policy with NOCACHE action, then the request bypasses all cache processing. This flag does not affect the processing of those requests that match an invalidation policy. The default is no value.

 © Copyright 2012 Citrix Systems, Inc.

verifyUsing Argument

Value settings for verifyUsing are:

- HOSTNAME
- HOSTNAME_AND_IP
- DNS

If the value of this attribute is set to HOSTNAME, then the URL, host name, and host port values in the incoming HTTP-request header must match before a cached object can be served. The IP address and the TCP port of the destination host are not matched. For certain deployments, the HOSTNAME setting can be a security risk.

The DNS parameter performs DNS resolution on the host and IP address in the request URL. This parameter offers an added security check and ensures that the request is from a valid source.

A rogue client can access a rogue server through the integrated cache using the following HTTP request:

```
GET / HTTP/1.1
Host: sensitive.foo.com
```

In this example, the integrated cache stores the rogue page served by the rogue server. Any subsequent client trying to access the root page from sensitive.foo.com is served the rogue page.

© Copyright 2012 Citrix Systems, Inc.

HOSTNAME Setting

The HOSTNAME setting should be set only if it is certain that no rogue client can access a rogue server through the integrated cache. The YES setting can lead to more hits if DNS-based load balancing is in use, and the same content can be served by multiple servers.

If the attribute is set to HOSTNAME_AND_IP, then the URL, host name, host port in the incoming HTTP-request header, IP address, and TCP port of the destination server must match.

If the set-cache-argument attribute is set to DNS, then the URL, host name, host port in the incoming HTTP request, and the TCP port should match. In regards to the IP address of the destination server, the hostname is used to perform a DNS lookup, and it is compared with the set of addresses returned by the DNS lookup. The default value is DNS, with a DNS default of no value.

 © Copyright 2012 Citrix Systems, Inc.

Caching Management

The objects in cache can be invalidated two ways with the command-line interface:

- Expiring an object ensures that the object is not served by integrated caching without re-validating it by the origin. Objects that do not carry validators, such as Etag or last-modified headers, cannot be revalidated. For these objects, flushing or expiring has the same effect.
- Flushing an object ensures that the object is never served by integrated caching again.

Invalidations are carried out at different levels of granularity. If the object is part of a non-parameterized content group, then the URL must match byte by byte with the stored URL. If the object is part of a parameterized content group, then the group name has to be specified and the URL must match the stored URL. The hostname and port must be the same strings that were present in the host-HTTP-request header of the cached object. If a port is not specified, then port 80 is used.

AppExpert Templates

AppExpert templates, which are reusable bundles of configuration information, provide an easy way to configure the NetScaler system to optimize the traffic of specific applications.

AppExpert templates require the application expert to categorize the workflow of the application into units. Then rules are planned for each of these units before you can start deploying the application using the NetScaler system.

The application view also provides a single interface from which you can manage in a few clicks all of the application-specific settings.

An additional benefit of being able to import and export templates is the ability for you to exchange these templates. Within organizations and between organizations, you and application experts can share optimizations and recommendations.

The terminology used when discussing AppExpert templates is more commonly used with applications. The interface provides an executive view of configuration and provides a single location to configure most of the NetScaler features.

Using AppExpert Templates

AppExpert templates are accessible from the applications section of the AppExpert node within the configuration utility.

You can use AppExpert templates to:

- Configure the various application features that the NetScaler system is optimizing.
- View which NetScaler functional modules, such as compression, caching, and Application Firewall, are optimized and active for a given application unit.

AppExpert templates can be downloaded, imported, modified, and exported. For more information about AppExpert templates and to download specific AppExpert templates built or certified by Citrix, see the AppExpert community web site at: *http://community.citrix.com/display/ns/AppExpert+Templates*.

These templates are easily imported into any NetScaler system running NetScaler 9.0 or higher, which allow you to quickly start the configuration and deployment process. Templates developed in-house can be easily exported and shared within the organization, or posted back to the AppExpert community web site.

The public endpoints and back-end service are two crucial pieces of information when configuring AppExpert templates.

Using the Application Interface

The interface for AppExpert templates allows you to see the following information by application component and NetScaler module:

 © Copyright 2012 Citrix Systems, Inc.

- Which individual NetScaler features are configured for an application
- Which features are inactive but available for configuration for an application

You can use the configuration utility to create and activate policies in each feature or deactivate individual policies for an application. Available activities related to AppExpert templates include:

- Downloading an application template from the AppExpert community web site
- Importing an AppExpert template
- Creating an application
- Creating application units
- Configuring back-end services
- Configuring public endpoints
- Modifying the expression of an application unit
- Changing the order of application units
- Applying features
- Exporting an application as a template
- Viewing statistics
- Viewing the visualizer

Applications Pane

AppExpert templates, not to be confused with entity templates, are found under the AppExpert node.

You can perform the following tasks in the Applications pane for the NetScaler system:

- Add or remove a new application on the NetScaler system.
- Remove an application from the NetScaler system.
- Configure public endpoints.
- Configure back-end services.
- Configure application firewall.
- Enable or disable EdgeSight Monitoring
- Import an AppExpert template to be used with the NetScaler system.
- Export an application into an AppExpert template configuration to use on other NetScaler systems.
- View statistics for the application or corresponding application units.
- Use the visualizer to view a graphical representation of the application.

© Copyright 2012 Citrix Systems, Inc.

Application Units

Application units describe the content that you may want to optimize. The following tasks can be performed for application units:

- Add a new application unit to the application.
- Configure policies for compression, caching, rewrite, content filtering, responder, and Application Firewall.

AppExpert Template Scalability

You can design a unique application configuration, export it to an application template, and then import it on other NetScaler systems.

An exported template has the ".gz" file extension.

Planning for Applications

Before creating and configuring a NetScaler system for an application, you need to identify the workflows of interest. Key terminology to guide the planning process includes application workflows, also known as application units within the configuration utility. These terms are used to define logically how the NetScaler system interfaces with a specific application.

The creation and use of AppExpert templates can be divided into two major tasks. Both tasks can be carried out by the same person or team in an organization.

1. Create the AppExpert template and application units
2. Import the AppExpert template

The first task is performed by an application expert, who knows the intricacies of the application, its workflows, and the optimizations that would benefit the application. This knowledge is used to create the AppExpert template and its application units (workflows) and to configure optimization and security policies for the application. The resulting template is exported providing a reusable AppExpert template that you can import whenever a similar application is deployed in the environment.

The second task is performed by an administrator who is responsible for deploying the NetScaler configuration of an application. The administrator imports the existing AppExpert template to the NetScaler system and provides the deployment unique to the environment, such as the IP addresses and ports of public endpoints and back-end services. This import results in a new completely configured application that appears on the system. You can, if required, further modify the policies or application units of the created application.

 © Copyright 2012 Citrix Systems, Inc.

Terminology

Application units define a subset of traffic for an application for which you are interested in creating policies. Examples include:

- Reports
- Documents
- Images
- Stylesheets
- Web services
- Portal pages

The definition of application units should be request based because the expressions are built upon request-based rules.

© Copyright 2012 Citrix Systems, Inc.

Methodology

The methodology behind deploying a NetScaler configuration for an application requires two steps:

1. Identification of application workflows of interest according to the type of content they generate from server to client
2. Creation of application units based on the identified workflows, defining specific rules that allow the NetScaler system to recognize each workflow

During the process of creating an application in the NetScaler system, you can enter the expression of the rule to identify each application unit (workflow). You can configure policies for features, such as compression, caching, rewrite, filtering, responder, and Application Firewall, that pertain to this application.

What is chosen as an application unit by the application expert depends on what needs to be optimized. For instance, if the application expert determines that compressing all Word documents will benefit the application, then the expert can create an application unit called "Word documents" with a rule that identifies the requests of this workflow (for example URL contains “/*.doc”).

It is beneficial to identify application units by which one can configure multiple features. This action streamlines the application definition by reducing the number of application units needed in an application.

Identifying and planning the workflows of an application, and the optimization policies that can be applied to each, is the main task of the application expert and requires knowledge of the application in order to optimize benefits of the feature.

 © Copyright 2012 Citrix Systems, Inc.

Application Example: Part 1

The following table provides an example of the SharePoint application broken down by workflow, the workflow characterizations, and the components of the workflow.

The workflows are broken into general categories, such as document management and portal management. The content is identified by specific file types, such as ".doc" and ".asp" file types, in the components column. These file types can be used to build application units.

Workflow	Characterized By	Components
Document management	Document sharing and storage, Microsoft Office documents, reports, spreadsheets, forms	.doc, .docx, .ppt, .pptx, .dot, .dotx, .docm, .dotm, .rtf, .txt, .wps, .pot, .potx, .pptm, .potm, .thmx, .ppsx, .ppsm, .pps, .ppam, .pdf, .csv, .prn, .xsn, .xls, .xslx, .xlt, .xltx, .xlsb, .xlsm, .xltm, .dif, .slk, .xlam, .xla
Styles and scripts	Stylesheets, Javascript	.css, .js
Image management	Images	.gif, .jpg, .jpeg, .tif, .tiff, .bmp, .wmf, .emf, .png
Portal management	ASP.NET, web pages	.asp, .aspx, .htm, .html, .mht, .mhtml, .xhtml, .xml, .jsp, .jspx
Web services definitions	WSDLs and WSILs	?wsdl, .wsdl, ?wsil, .wsil, .xml
Web services schema	XSDs	.xsd

© Copyright 2012 Citrix Systems, Inc.

Application Example: Part 2

The table illustrates how you can take this example one step further. In this example, after identifying the content workflows or pertinent content, a NetScaler administrator determines what policies can be applied to optimize the application. Some of this determination is based on general NetScaler recommendations, some is based on application-specific recommendations, and some is based on internal organizational preferences.

For example:

- Documents can be compressed and cached.
- Images can be cached but not compressed.
- Portal pages should not be compressed or cached.

Every organization has unique philosophies on what content can or should be cached, compressed, and configured for optimization. You should coordinate with the appropriate groups before configuring an application.

SharePoint	Compression	Caching	Rewrite	Filter	Responder
Documents	X	X			
Styles and scripts	X	X			
Images		X			
Portals					
Web services	X				
Web service schema	X	X			
Default	X	X			

© Copyright 2012 Citrix Systems, Inc.

AppExpert Template Deployment

To deploy an application template:

1. Configure the back-end services and public endpoints.
2. Create application units.
3. Apply the desired features.

The order of the tasks necessary for deploying an application varies depending on whether the application is generic, such as Microsoft SharePoint, or custom.

For many of the generic applications, you can download a template that includes the majority of the generic optimizations. From there, you import the template, configure the public endpoints and back-end services, and make any desired customizations to the feature rules.

For custom applications, a template might have been prepared by a local application expert. If this is the case, you can import that template and configure the public endpoints and back-end services.

> Even when you create and configure an application directly on a NetScaler system (without using an existing template), a good practice is to export the application once configured into an AppExpert template. Use that template when configuring the same application on other NetScaler systems, instead of rebuilding the application configuration on each system.

Test Your Knowledge

1. Every cached object is made a member of a _____. (Choose the correct answer.)
    a. content group
    b. built-in policy
    c. dynamic response
    d. compression action
2. Which option prevents an object in a content group from being flushed due to memory pressure?
    a. Memory limit
    b. Pinned
    c. Integrated Caching
    d. Flash Cache
3. Which feature causes the NetScaler system to allow only one request to reach the server while other requests are queued until the server responds?
    a. Compression
    b. Flash Cache
    c. AppExpert
    d. Integrated Caching
4. When you flush an object, you ensure that the object _____. (Choose the correct answer.)
    a. never expires
    b. is never pinned
    c. is invalidated
    d. is never served by integrated caching again

 © Copyright 2012 Citrix Systems, Inc.

Module 13

Monitoring

© Copyright 2012 Citrix Systems, Inc.

Overview

There are various methods for monitoring a NetScaler deployment. SNMP allows you to generate traps for abnormal conditions on the NetScaler system. Citrix AppFlow is an open standards technology that transforms the data collected by existing networking devices into powerful operational and business intelligence. And EdgeSight Monitoring for NetScaler monitors the end-user experience with web applications that are served in an environment using NetScaler. In this module, you will also learn about audit server logging, the reporting tool, Command Center, and the Dashboard.

After completing this module, you will be able to:

- Determine which data to monitor in the environment.
- Monitor the NetScaler system using the syslogs.
- Monitor the NetScaler system by viewing statistics through the Dashboard.
- Assess the health of the NetScaler system by reviewing built-in reports.
- Configure monitoring with SNMPv1 and SNMPv2 on the NetScaler system.
- Explain how AppFlow provides visibility to application behavior, performance, and security.

© Copyright 2012 Citrix Systems, Inc.

Monitoring Needs

The NetScaler system provides integrated, real-time monitoring of application and client traffic, along with the capability to feed back critical application delivery information to automatically optimize the system and load-balancing policies.

The following use cases are examples of the growing monitoring and information demands:

- Mapping the end-user experience for e-commerce sales
- Ensuring improved load balancing across the datacenter
- Tracking traffic through web applications
- Web application performance
- Identifying when application response times exceed service level agreements (SLAs) for transactions
- Monitoring the end-user experience
- Monitoring abnormal client-server behavior based on packet counts, byte counts, and server response times

 © Copyright 2012 Citrix Systems, Inc.

Monitoring Methods

A NetScaler system provides several ways to monitor your environment, including:

- SNMP
- AppFlow
- EdgeSight Monitoring
- Action Analytics
- Command Center
- Reporting tool
- Dashboard

SNMP

You can use Simple Network Management Protocol (SNMP) to configure the SNMP agent on the NetScaler system to generate asynchronous events, which are called traps. The traps are generated whenever there are abnormal conditions on the NetScaler system. The traps are then sent to a remote device called a trap listener, which sends a signal that there is an abnormal condition on the NetScaler system. Or, you can query the SNMP agent for system-specific information from a remote device called an SNMP management system. The agent then searches the management information base (MIB) for the data requested and sends the data to the SNMP management system.

AppFlow

AppFlow provides visibility at the transaction level for HTTP, SSL, TCP, and SSL_TCP flows. You can sample and filter the flow types that you want to monitor. AppFlow uses actions and policies to send records for a selected flow to a specific set of collectors. An AppFlow action specifies which set of collectors will receive the AppFlow records. Policies, which are based on advanced expressions, can be configured to select flows for which flow records will be sent to the collectors specified by the associated AppFlow action.

EdgeSight Monitoring

The EdgeSight Monitoring application provides web page monitoring data with which you can monitor the performance of various web applications served in a NetScaler environment. You can now export this data to AppFlow collectors to get an in-depth analysis of the web page applications. AppFlow, which is based on IPFIX standard, provides more specific information about web application performance than does EdgeSight Monitoring alone. You can configure both load-balancing and content-switching virtual servers to export EdgeSight Monitoring data to AppFlow collectors.

© Copyright 2012 Citrix Systems, Inc.

Action Analytics

NetScaler Action Analytics collects detailed information on administrator-selected application traffic flows and generates multidimensional views of what is happening in real-time. Graphical and tabular displays enable you to instantly view critical application-delivery parameters, such as which top URLs are being accessed or which top clients are consuming application resources, and then drill down for more detailed statistics including number of overall requests, total bandwidth being consumed, and the response times of the back-end infrastructure delivering the application content.

Command Center

Citrix Command Center is a management and monitoring solution for Citrix application networking products, which includes Citrix NetScaler. Command Center enables network administrators and operations teams to manage, monitor, and troubleshoot the entire global application delivery-infrastructure from a single, unified console. This centralized management solution simplifies operations by providing you with real-time performance monitoring of your enterprise-wide application-delivery infrastructure and automating management tasks that need to be executed across multiple devices.

The following list includes some of the management tasks that are simplified by using Command Center:

- Quickly address and resolve device and network issues and keep the network running effectively by monitoring and managing the SNMP and syslog events generated on your devices.
- Understand the traffic patterns, gather data for capacity planning, and monitor the performance of the entire application delivery infrastructure by using historical charts and performance graphs.
- Monitor and manage the states of virtual servers, services, and service groups across the NetScaler infrastructure.
- Troubleshoot configuration errors or recover unsaved configuration on sudden system shutdown by running audit policies.

Available as a software platform or a hardware appliance, Command Center enables you to monitor and control a diversified deployment of multiple NetScaler solutions including:

- NetScaler ADC
- NetScaler SDX
- NetScaler Access Gateway
- NetScaler Application Firewall
- Branch Repeater
- CloudBridge
- XenServer

 © Copyright 2012 Citrix Systems, Inc.

The NetScaler Command Center hardware appliance can manage 300 Citrix networking devices, includes 500GB of built-in storage, and is preloaded with a database.

To monitor and manage Citrix devices, you need to connect to the Command Center server by using the HTML Web client, and then add the devices for discovery. For more information about Citrix Command Center, see Citrix eDocs at *http://edocs.citrix.com*.

Reporting Tool

The NetScaler system provides the reporting tool in order to parse and display the data contained in the syslog file. The reporting tool is a web-based interface that provides built-in reports that display statistics collected by the nscollect utility.

Reports allow you to plot and monitor statistics for the various functional groups over a specified time interval, which assists in the troubleshooting and analyzing of the behavior of the NetScaler system. You can access the reporting tool by selecting **Reporting** when logged on to the NetScaler system.

Working with Reports

You can plot and monitor statistics on the NetScaler system over a specified time interval. Reports enable you to troubleshoot or analyze the behavior of your system. There are two types of reports: built-in reports and custom reports. The content for built-in or custom reports can be viewed in a graphical format or a tabular format. The graphical view consists of line, area, and bar charts that can display up to 32 sets of data, also known as counters. The tabular view displays the data in columns and rows. This view is useful for debugging error counters.

The default report that is displayed in the reporting tool is CPU versus memory usage and HTTP requests rate. Reports can be generated for the last hour, last day, last week, last month, last year, or a customized interval of time.

You can do the following with reports:

- Change the graphical display type, such as bar chart or line chart.
- Customize charts in a report.
- Export the chart as an Excel comma-separated value (CSV) file.
- View the charts in detail by zooming in, zooming out, or using a drag-and-drop operation (scrolling).
- Set a report as the default report for viewing whenever you log on.
- Add or remove counters.
- Refresh reports to view the latest performance data.

Using Built-in Reports

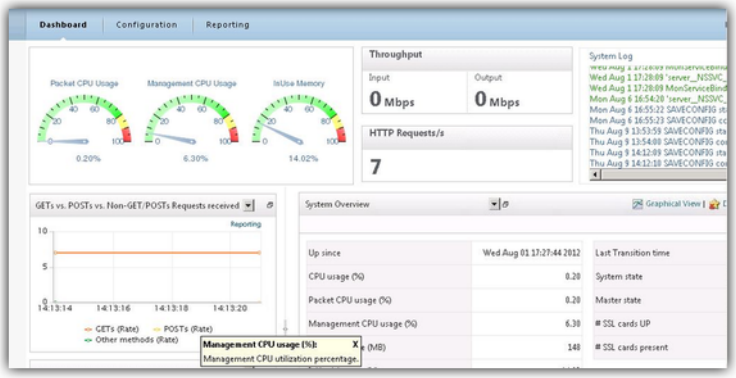
The reporting tool provides built-in reports for frequently viewed data. Built-in reports are available for the following seven functional groups:

- System
- Network
- SSL
- Compression
- Integrated Cache
- Access Gateway
- Application Firewall

By default, the built-in reports are displayed for the last day. However, you can view the reports for the last hour, last week, last month, or last year.

> You cannot save changes to built-in reports, but you can save any modifications to a built-in report by saving it as a custom report.

Dashboard



The monitoring page in the configuration utility has been enhanced and renamed the Dashboard. The new Dashboard is HTML-based and replaces the Java-based Dashboard. The Dashboard displays critical performance statistics and provides real-time data. The data displayed is for the last five minutes of operation and is updated every seven seconds.

System and Feature Counters

The Dashboard monitors both system and features counters.

 © Copyright 2012 Citrix Systems, Inc.

System counters include:

- CPU utilization
- Memory utilization
- System throughput
- System statistics
- Protocol statistics
- High availability
- Bridge
- ACL
- Network entities
- SNMP
- SSL
- System health

Feature counters include:

- Audit
- AAA
- VPN
- TCP
- Load balancing
- Content switching
- Services
- HTTP compression
- Decompression
- Integrated cache

Certain features are dependent on the enabled licenses.

Dashboard Components

The Dashboard components give you information on the health and status of the NetScaler system.

CPU Utilization Pane	The CPU Utilization pane displays the real-time CPU utilization of the system as a percentage of the total capacity. The color codes for the pane are:

- Green, which indicates that the CPU utilization is within safe limits
- Yellow, which indicates that the CPU utilization is high
- Red, which indicates that the CPU utilization has reached a critical limit

Memory Utilization Pane

The Memory Utilization pane displays the real-time memory utilization of the system. While the dial indicates the percentage of the memory used, you can drag the mouse cursor over the pane for an absolute value. The color codes for the pane are:

- Green, which indicates that the memory utilization is within safe limits
- Yellow, which indicates that the memory utilization is high
- Red, which indicates that the memory utilization has reached a critical limit

System Throughput Pane

The System Throughput pane displays throughput in terms of incoming and outgoing traffic.

HTTP Requests per Second Pane

The HTTP Request per Seconds pane illustrates the HTTP requests served by the system each second.

System Log Pane

The System Log pane provides a view of system events and alerts that are generated when the Dashboard is connected to the system. The data is real time and has an unlimited history; however, all data is lost when the Dashboard is restarted.

Built-in Comparative Chart Pane

The Built-in Comparative Charts pane monitors a built-in set of counters using charts. The charts depict the variation of two or more counters over time. Options for viewing the charts include chart type, chart appearance, legends, independent window views, and custom charts.

Group Monitoring Pane

The Group Monitoring pane provides a view of all counters of a feature or protocol and the ability to plot any number of counters on custom charts. By default, this pane provides a snapshot of the system.

 © Copyright 2012 Citrix Systems, Inc.

NetScaler Log Management

Newnslog stores console messages, events, and performance statistics, which are written to newnslog every seven seconds. Every three hours the log is saved. Once saved, these newsnslog files roll on a periodic basis based on a specified lifetime of the nsconmsg process. When a log rolls, each older log file is compressed using GZIP and saved with a number from 0 through 100. Newnslog is a binary file stored in `/var/nslog`.

The Manage Logs option in the Diagnostic pane provides the ability to view information from the log files, save viewed data, download the log files, and delete the unwanted log files from the NetScaler system. The following actions can be performed:

- Viewing events
- Viewing log files duration
- Viewing events from a specific time
- Viewing console messages
- Downloading log files
- Deleting log files

Audit Logging

Audit logging enables you to log the NetScaler states and status information collected by various modules in the kernel and in the user-level daemons. The NetScaler system allows you to customize the logging of system events according to the need of a site. You can record events either to files on the NetScaler system or to external log servers. The NetScaler system supports two logging formats:

- syslog
- nslog

By default, the NetScaler system records system events in syslog format to `/var/log/ns.log` using local0 and records SSL VPN events to `/var/log/nsvpn.log` using local1. Logging of these events is enabled by default. Syslog uses UDP over port 514, and syslog traffic is sent in clear text. The NetScaler kernel controls syslog processing. Nslog is a proprietary binary format that records more detailed event information than the syslog format. You can log syslog and nslog events to either a local file on the NetScaler system or to a remote server.

`Syslog.conf` should not be modified.

Configuring NetScaler for Audit Logging

Policies define the syslog or nslog protocol and the server actions define where each log is sent. For server actions, you specify the system information, which runs the syslog or the nslog server.

The NetScaler system logs the following information related to TCP connections:

- Source port
- Destination port
- Source IP
- Destination IP
- Number of bytes transmitted and received
- Time period for which the connection is open

To configure audit logging on the NetScaler system, you must:

1. Configure audit servers
2. Configure audit policies
3. Bind the audit policies globally

Configuring an Auditing Server

An external nslog audit server can run either the Linux, FreeBSD, or Windows operating system.

> Local logging still occurs on the NetScaler system when an external audit server is configured.

You can configure an audit server on the NetScaler system by specifying the following parameters:

- The name of the syslog server action or nslog server action
- IP address of the auditing server
- Port through which to communicate
- Severity levels of messages to be logged
- Format of the date stamp
- Facility value (RFC 3164) assigned to the log message
- Time zone for the time stamp
- TCP logging
- ACL logging
- Enable user-configurable log messages
- Enable or disable export log messages to the AppFlow collectors.

In the configuration utility, go to **Configuration** > **System** > **Auditing**. When using the command-line interface, enter:

 © Copyright 2012 Citrix Systems, Inc.

```
add audit logAction name ServerIPAddress -serverPort port -
loglevel LogLevel
 -dateFormat ( MMDDYYYY | DDMMYYYY ) -logFacility logFacility
 -tcp (None | ALL ) -timeZone ( GMT_TIME | LOCAL_TIME )
```

Replace logAction with the type, either syslogAction for a syslog auditing server, or nslogAction for an nslog auditing server.

Global Auditing Parameters

Global auditing parameters help you log all the states and status information collected by different modules in the kernel, as well as in the user-level daemons in the NetScaler system.

Configure global auditing parameters on the NetScaler system by specifying the following:

- Auditing type
- IP address of the external server used to store auditing messages
- Port number used for communication between the system and the external logging server
- Level of logging detail
- Date format
- Time zone
- Log facility value
- TCP logging

In the configuration utility, go to **Configuration** > **System** > **Auditing** > **Policies**. In the command-line interface, enter:

```
add audit Params name ServerIPAddress  -serverPort port -
logLevel LogLevel
 -dateFormat ( MMDDYYYY | DDMMYYYY ) -logFacility logFacility
 -tcp (None | ALL ) -timeZone ( GMT_TIME | LOCAL_TIME )
```

Replace Params with either syslogParams for a syslog auditing server, or nslogParams for an nslog auditing server.

Configuring Auditing Policies

Auditing policies decide which messages are generated and logged during the session. These messages are logged in the syslog or nslog format. Different types of messages are logged based on the level of logging selected.

Auditing policies are used for the Access Gateway feature of the NetScaler system.

You can add an auditing policy on the NetScaler system by specifying the policy name and rule action. In the configuration utility, go to **Configuration** > **System** > **Auditing** > **Policies**. Using the command-line interface, enter:

```
add audit policy Policy name rule action
```

Binding Auditing Policy

You can configure auditing on the NetScaler system at several levels in the following order of priority:

1. Global
2. Virtual server
3. Group
4. User

At each of these levels, you can define multiple auditing policies. You can either bind auditing policies globally or to virtual servers, groups, and users.

You can bind an auditing policy on the NetScaler system by specifying the policy and assigning a priority. In the configuration utility, go to **Configuration** > **System** > **Auditing** > **Policies**. When using the command-line interface, enter:

```
bind system global PolicyName -priority positive_integer
```

Audit Messages

You can view recent and historical audit messages in both the configuration utility and in the command-line interface.

You can view recent audit messages on the NetScaler system by specifying the level of detail to be displayed and the number of audit messages to be shown and the style to display the messages. In the configuration utility go to **Configuration** > **System** > **Auditing**. When using the command-line interface, enter:

```
show audit messages -logLevel logLevel -
numOfMesgs positive_integer
```

You can also view historical audit messages on the NetScaler system by going to **Configuration** > **System** > **Auditing** > **Syslog messages** in the configuration utility.

 © Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

1. Which component within SNMP is responsible for monitoring a group of devices on a network?
    a. Management system
    b. Agent
    c. Community string
    d. Managed device
2. Which tool can you use to manage, monitor, and troubleshoot your entire global application infrastructure from a single console?
    a. SNMP management system
    b. Dashboard
    c. Reporting Tool
    d. Citrix Command Center
3. Which port does syslog use by default?
    a. TCP port 80
    b. UDP port 443
    c. TCP port 22
    d. UDP port 514
4. How often is newnslog rotated?
    a. Every day
    b. Every two days
    c. Every week
    d. Every two weeks

© Copyright 2012 Citrix Systems, Inc.

Simple Network Management Protocol

Simple Network Management Protocol (SNMP) is an application-layer protocol that provides a method for network devices to communicate status and configuration information to a management system. By default, SNMP uses UDP on port 161 for queries and port 162 for traps.

You can do the following SNMP configurations on the NetScaler system:

- Specify system information that is displayed using the NetScaler SNMP Management Information Base (MIB).
- Specify SNMP traps that track various parameters, such as usage and interface status.

SNMP consists of the following components:

- Managed devices
- Agents
- Management systems
- MIBs
- Community strings
- Traps

A NetScaler system supports SNMPv1, SNMPv2, and SNMPv3. SNMPv1 and SNMPv2 use a simple and non-secure authentication feature called community strings. SNMPv3 adds security with authentication, privacy, and access control. SNMPv3 is not a replacement for previous versions; it defines a security capability to be used in conjunction with SNMPv1 or SNMPv2.

SNMPv3

Simple Network Management Protocol version 3 (SNMPv3) is based on the structure and architecture of SNMPv1 and SNMPv2. However, SNMPv3 enhances the basic architecture to incorporate administration and security capabilities, such as authentication, access control, data integrity checks, data origin verification, message-timeliness checks, and data confidentiality.

SNMPv3 configuration packets (views, groups, users, and engineID) are sent to the SNMP daemon on the same UDP port 161 with source number IP address and destination number IP address of the loopback address (127.0.0.1) in order to distinguish them from protocol data units.

To implement message-level security and access control, SNMPv3 introduces the user-based security model (USM) and the view-based access control model (VACM).

User-based Security Model (USM)	USM provides message-level security. It enables you to configure users and security parameters for the SNMP agent and the SNMP.

USM offers the following features:

 © Copyright 2012 Citrix Systems, Inc.

- Data integrity: To protect messages from being modified during transmission through the network.
- Data-origin verification: To authenticate the user who sent the message request.
- Message timeliness: To protect against message delays or replays.
- Data confidentiality: To protect the content of messages from being disclosed to unauthorized entities or people.

View-based Access Control Model (VACM)	VACM enables you to configure access rights to a specific subtree of the MIB based on various parameters, such as security level, security model, user name, and view type. It enables you to configure agents in order to provide different levels of access to the MIB to different managers.

SNMPv3 Entities

The NetScaler system supports the following entities that enable you to implement the security features of SNMPv3:

SNMP engines	SNMP engines are service providers that reside in the SNMP agent. They provide services such as sending, receiving, and authenticating messages. SNMP engines are uniquely identified using engine IDs. The NetScaler has a unique engineID based on the MAC address of one of its interfaces. It is not necessary to override the engineID. However, if you want to change the engine ID, you can reset it.
SNMP views	SNMP views restrict user access to specific portions of the MIB. SNMP views are used to implement access control.
SNMP groups	SNMP groups are logical aggregations of SNMP users. They are used to implement access control and to define the security levels. You can configure an SNMP group to set access rights for users assigned to that group, thereby restricting the users to specific views.
SNMP users	SNMP users are the SNMP management systems that the agents allow to access the MIBs. Each SNMP user is assigned to an SNMP group. You need to configure users at the agent and assign each user to a group.

These entities function together to implement the SNMPv3 security features. Views are created to allow access to subtrees of the MIB. Then, groups are created with the required security level and access to the defined views. Finally, users are created and assigned to the groups.

> The view, group, and user configuration are synchronized and propagated to the secondary node in a high-availability pair. However, the engine ID is neither propagated nor synchronized as it is unique to each NetScaler system.

Managed Devices

Managed devices are network nodes that reside in a managed network. Managed devices collect and store management information and make this information available to a network management system (NMS) using SNMP. Managed devices are sometimes called network elements and can be routers, access servers, switches and bridges, hubs, computer hosts, or printers. The NetScaler system is an example of a managed device.

SNMP Agents

Agents are software modules that reside on managed devices. Agents listen on port 161 for request packets from managers and respond to requests for information and actions. In addition to the protocol and application layers, agents must also communicate with the operating system kernel.

Typically, SNMP uses UDP port 161 for the agent and port 162 for the manager communications. The manager may send requests from any available port (source port) to port 161 for the agent (destination port). The agent response will be sent back to the source port, and the manager will receive traps on port 162. The agent may generate traps from any available port.

The SNMP agent on a NetScaler system supports SNMPv1, SNMPv2, and SNMPv3.

> The NetScaler system allows traps to be sent to a different port other than 162, if needed.

SNMP Management Systems

Management systems are devices that poll agents for information. Management systems can do a passive statistics-gathering function, or they can attempt to manage the network actively by setting new values in read-write variables on some hosts.

SNMPv1 management systems (programs on other servers that request SNMP information from the NetScaler system) use the NS-MIB-smiv1.mib file when processing SNMP queries. The NS-MIB-smiv2.mib file is used by SNMPv2 and SNMPv3 management systems and by SNMPv2 trap listeners.

Management systems tested with the NetScaler system include:

 © Copyright 2012 Citrix Systems, Inc.

- HP OpenView
- WhatsUp Gold
- Net-SNMP (Linux)
- MGSoft (Windows)

Management Information Base

A management information base (MIB) is a collection of objects. Each object in a MIB has a unique object identifier (OID) and represents a piece of information from an agent. A MIB is organized hierarchically. A new MIB is defined for each new device.

The NetScaler system has the standard MIB and a private MIB for NetScaler-specific data. The standard MIB has items such as ifOperStatus and ifInOctets, and the private MIB has items such as virtual server current connections and total requests/responses.

The enterprise-registered OID for the NetScaler system is 1.3.6.1.4.1.5951.4 (5951).

Community Strings

A community string is a type of authentication that allows access to managed devices. The management system and agent must be in the same community.

If a management system sends a request with the correct community string, the device responds with the requested information in accordance with the allowed level of access for that community string. If the community string is incorrect, the device simply discards the request and does not respond.

Community string types include:

- Read-only community string
- Read-write community string
- Trap community string

The default community string on the NetScaler system is public. You should change the default community string to read-only, read-write, or trap.

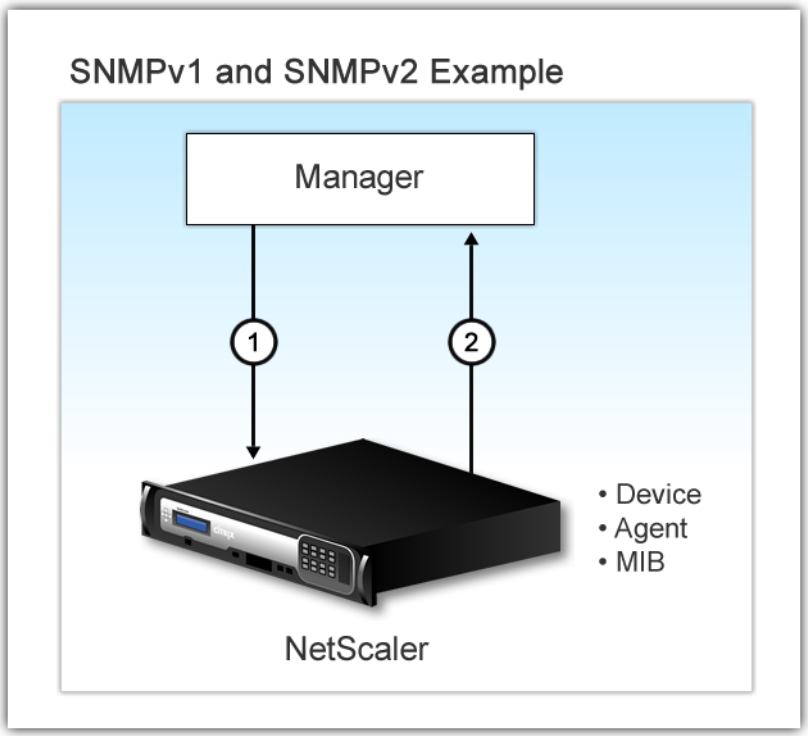
Traps

You can configure the NetScaler system to generate asynchronous events, which are called traps. A trap is generated when there is an abnormal condition, or a specified threshold is reached on the NetScaler system. The traps are sent to a remote device called a trap listener. The traps and trap listener help you monitor the NetScaler system and respond promptly to any issues.

The NetScaler system provides a set of condition entities called SNMP alarms. When the condition in any SNMP alarm is met, the NetScaler system generates SNMP trap messages that are sent to the configured trap listeners. For example, when the LOGIN-FAILURE alarm is enabled, a trap message is generated and sent to the trap listener when a logon failure occurs on the NetScaler system. To configure the NetScaler to generate traps, you need to enable and configure alarms. Then, you specify trap listeners to which the NetScaler system will send the generated trap messages.

All SNMP traps are logged in syslog. The NetScaler system supports SNMPv1 and SNMPv2 traps, but not SNMPv3 traps.

SNMPv1 and SNMPv2 Communication Process



The figure and the following process provide an overview of the SNMPv1 and SNMPv2 communication process occurring between an SNMP management system and the NetScaler system, which is the managed device running the SNMP agent and housing the MIB.

1. The management system makes a GET request to the NetScaler system to retrieve data.

 © Copyright 2012 Citrix Systems, Inc.

2. The NetScaler system validates the request by evaluating the community string and source IP address of the management system.

You can configure up to 100 management systems on a NetScaler system. If you do not configure an SNMP management system, the NetScaler system accepts and responds to SNMP queries from all IP addresses on the network. If you configure one or more SNMP management systems, then the NetScaler system accepts and responds to only SNMP queries from those specific IP addresses.

To Configure SNMPv1 and SNMPv2

1. Configure a management system by specifying the IP address and subnet mask. In the configuration utility, go to **Configuration** > **System** > **SNMP** > **Managers** > **Add** > **Create SNMP Manager**.

   When using the command-line interface, enter:

   ```
   add snmp manager IPAddress -netmask netmask
   ```

2. Set access privileges for the network management application end user by configuring the community string. In the configuration utility, go to **Configuration** > **System** > **SNMPCommunity**.

   When using the command-line interface, enter:

   ```
   add snmp community communityName permissions
   ```

3. Set the MIB variable by specifying the contact, name, location, and custom ID. In the configuration utility, go to: **Configuration** > **System** > **SNMP** > **Configure SNMP MIB**.

   When using the command-line interface, enter:

   ```
   add snmp mib -contact string -name string  -location string -
   customID string
   ```

4. Configure a trap. In the configuration utility, go to **Configuration** > **System** > **SNMP** > **Traps**.

   When using the command-line interface, enter:

   ```
   add snmp trap trapClass trapDestination -version (V1|V2)  -
   destPortport  -communityName string  -srcIPip_addr  -
   severityseverity
   ```

5. Set a trap threshold. Configure an alarm by enabling the alarm and specifying the severity level and current state of the alarm. In the configuration utility, go to **Configuration** > **System** > **SNMP** > **Alarms**.

   When using the command-line interface, enter:

   ```
   set snmp alarm alarm Name -thresholdValue positive_integer -
   normalValue positive_integer -time secs -
   state  ( ENABLED | DISABLED ) -severity severity -
   logging(ENABLED | DISABLED)
   ```

Configuring SNMP Management Systems

You can configure up to 100 management systems on a NetScaler system. For an IPv4 SNMP management system, you can specify a host name instead of using the IP address of the manager. If

 © Copyright 2012 Citrix Systems, Inc.

you do so, you must add a DNS name server that resolves the host name of the SNMP management system to its IP address. You can add up to five host-name-based SNMP managers.

> The NetScaler system does not support use of host names for SNMP management systems that have IPv6 addresses. You must specify the IPv6 address.

Add a management system on the NetScaler system by specifying the IP address and subnet mask. Using the configuration utility, go to **Configuration** > **System** > **SNMP** > **Managers**. Using the command-line interface, enter:

```
add snmp manager IPAddress -netmask netmask
```

You can view details for the management system on the NetScaler system by using the following command:

```
show snmp manager
```

Configuring MIB System Variables

A management information base (MIB) represents the set of values that can be requested from a particular agent. Each network device may have its own MIB that includes the types of information appropriate to the device. For example, a firewall has different values than a layer-2 switch.

Show and set the SNMP MIB on the NetScaler system by specifying the contact person, system name, physical location of the system, and the custom ID for the system. In the configuration utility, go to **Configuration** > **System** > **SNMP** > **Configure SNMP MIB**. Using the command-line interface, enter:

```
set snmp mib -contact string  -name string -customID string
```

Configuring Community Strings

Community strings are passwords used to authenticate SNMP queries from SNMP management systems. For example, if you associate two community strings, such as abc and bcd, to the query type GET NEXT, the SNMP agent on the NetScaler system considers only those GET NEXT SNMP query packets that contain abc or bcd as the community string.

You can configure a maximum of 20 SNMP communities on the NetScaler system. Each community string can include up to 32 characters. The permissions are:

- GET, which receives a piece of management information
- GET NEXT, which retrieves sequences of management information
- GET BULK, which is a faster way to receive sequences of management information

- ALL, which enables all access permissions

You can add a community string on the NetScaler system by specifying the community name and setting the permission values. In the configuration utility, go to **Configuration** > **System** > **SNMP** > **Community**. If you are using the command-line interface, enter:

```
add snmp community communityName permissions
```

Configuring Alarms

The NetScaler system provides a set of condition entities called SNMP alarms. When the condition set for an SNMP alarm is met, the NetScaler system generates SNMP trap messages that are sent to the configured trap listeners.

Configuring an alarm consists of enabling the alarm and setting the severity level at which a trap is generated. The following are the severity levels, defined in the NetScaler system, in decreasing order of severity:

1. Critical
2. Major
3. Minor
4. Warning
5. Informational

After an alert trap is sent, if the counter drops to the normal level or below, a second trap is sent. If more than 10 authentication traps are generated in a span of twenty seconds, the NetScaler system does not generate additional authentication traps for the next 60 seconds. This behavior is useful when the system is under an SNMP attack.

You can configure an alarm on the NetScaler system by enabling the alarm and specifying the severity level. In the configuration utility, go to **Configuration** > **System** > **SNMP** > **Alarms**. When using the command-line interface, enter:

```
set snmp alarm alarm name -thresholdValue positive_integer -
normalValue positive_integer -time secs -
state ( ENABLED | DISABLED )  -severity severity  -
logging ( ENABLED | DISABLED )
```

Configuring SNMP Traps

After configuring the alarms, you need to specify the trap listener to which the NetScaler system sends the trap messages. Apart from specifying parameters such as IP or IPv6 address and the destination port of the trap listener, you can specify the type of trap, and the SNMP version. You can configure two types of traps on the NetScaler system:

 © Copyright 2012 Citrix Systems, Inc.

- Generic SNMP traps (standard to all SNMP-enabled devices)
- NetScaler-specific traps

You can configure a maximum of 20 trap listeners for receiving either generic or specific traps. Each of these trap types can be enabled and sent to different destinations on a configurable port. In addition, you can now enable the NetScaler system to log any SNMP trap messages even when no trap listeners are specified on the system. By default, the system logs any SNMP trap messages when at least one trap listener is specified.

To add an SNMP trap on the NetScaler system using the configuration utility, go to **Configuration** > **System** > **SNMP** > **Traps**. To add an SNMP trap using the command-line interface, enter:

```
add snmp trap trapClass trapDestination -version (V1|V2) -
destPort port -communityName string -srcIP ip_addre -
severity severity
```

Test Your Knowledge

1. Which tool can you use to parse and display data contained in the syslog file?
    a. Reporting tool
    b. Dashboard
    c. Command Center
    d. EdgeSight Monitoring
2. Which entity is a type of authentication that allows access to managed devices?
    a. Trap
    b. Agent
    c. MIB
    d. Community string
3. What is the maximum number of SNMP managers that you can configure on a NetScaler system?
    a. 10
    b. 50
    c. 100
    d. 1000
4. Which are two enhancements that SNMPv3 provides to the SNMPv1 and SNMPv2 architecture? (Choose two.)
    a. Secure authentication
    b. MIBs
    c. Community strings
    d. Access control
5. What is the purpose of a community string?
    a. Authenticate SNMP queries from SNMP managers
    b. Represent the set of values that can be requested from an agent
    c. Generate asynchronous events
    d. Store console messages, events, and performance statistics

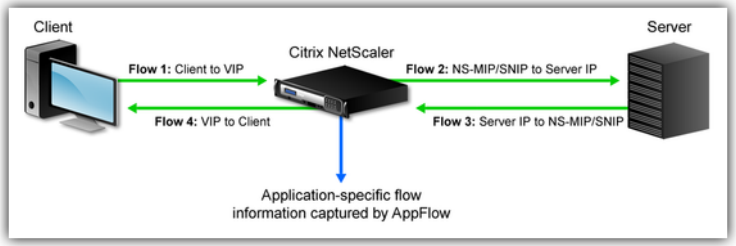
 © Copyright 2012 Citrix Systems, Inc.

AppFlow on the NetScaler System

The NetScaler system is a central point of control for all application traffic in the datacenter. It collects information about flow and about the end user's session that is valuable for application-performance monitoring, analytics, and business intelligence applications. It also collects web page-performance data and database information. AppFlow transmits the information by using the Internet Protocol Flow Information eXport (IPFIX) format. IPFIX is the standardized version of Cisco's NetFlow and is widely used to monitor network flow information. Using UDP as the transport protocol, AppFlow transmits the collected data, called flow records, to one or more IPv4 collectors. The collectors aggregate the flow records and generate real-time or historical reports. AppFlow provides visibility at the transaction level for HTTP, SSL, TCP, and SSL_TCP flows.

AppFlow uses actions and policies to send records for a selected flow to a specific set of collectors. An AppFlow action specifies which set of collectors will receive the AppFlow records. Policies, which are based on advanced expressions, can be configured to select flows for which flow records will be sent to the collectors specified by the associated AppFlow action. To limit the types of flows, you can enable AppFlow for a virtual server. AppFlow can also provide statistics for the virtual server. You can enable AppFlow for a specific service, representing an application server, and monitor the traffic to that application server.

How AppFlow Works



In a common deployment scenario, inbound traffic flows to a virtual IP address (VIP) on the NetScaler system and is load balanced to a server. Outbound traffic flows from the server to a mapped or subnet IP address on the NetScaler system and from the VIP to the client. A flow is a unidirectional collection of IP packets identified by the following five tuples: sourceIP, sourcePort, destIP, destPort, and protocol.

Configuring AppFlow

You can configure AppFlow in the same manner as most other policy-based features. The following provides an overview for configuring AppFlow:

1. Enable the AppFlow feature.
2. Specify the collectors to which the flow records are sent.

© Copyright 2012 Citrix Systems, Inc.

3. Define the actions. Actions are sets of configured collectors.
4. Configure one or more policies and associate an action to each policy. The policy tells the NetScaler system to select requests; the flow records of these requests are sent to the associated action.
5. Finally, bind each policy either globally or to a specific virtual server to put it into effect.

You can further set AppFlow parameters to specify the template-refresh interval and to enable the exporting of httpURL, httpCookie, and httpReferer information. On each collector, you must specify the NetScaler IP address as the address of the exporter.

The configuration utility provides tools that help users define the policies and actions which determine how the NetScaler system exports records for a particular flow to a set of collectors.

To enable AppFlow using the configuration utility, go to **Configuration** > **System** > **Settings**. In the details pane, go to **Modes and Features** > **Configure advanced features** and select the AppFlow check box.

To enable AppFlow using the command-line interface, enter:

```
enable ns feature appflow
```

Specifying an AppFlow Collector

A collector receives flow records generated by the NetScaler system. To be able to send flow records, you must specify at least one collector, but you can specify up to four.

If needed, you can remove unused collectors.

There are three parameters you must specify when creating a collector:

- Name
- IP address
- Port

To specify a collector using the configuration utility, go to the navigation pane **Configuration** > **System** > **AppFlow** > **Collectors** and click **Add**. Using the command-line interface, enter

```
add appflowCollector name  -IPAdress ipaddress -port port_number
```

AppFlow Action Configuration

An AppFlow action is a set of collectors to which the flow records are sent if the associated AppFlow policy matches. There are three parameters for configuring an AppFlow action:

 © Copyright 2012 Citrix Systems, Inc.

- Name
- Collectors
- Comment

When using the configuration utility, go to **Configuration** > **System** > **AppFlow** > **Actions** and then click **Add**. When using the command-line interface, enter

```
add appflow action name -collectors string -comment string
```

AppFlow Policy Configuration

After you configure an AppFlow action, you must configure an AppFlow policy. An AppFlow policy is based on a rule, which consists of one or more expressions. There are three parameters for configuring an AppFlow action:

- Name
- Expression
- Action

To configure an AppFlow action using the configuration utility, go to **Configuration** > **System** > **AppFlow** > **Policies** and then click **Add**. Using the command-line interface, enter:

```
add appflow policy name expression action
```

The configuration utility provides assistance that is not available at the command-line interface when creating and managing AppFlow policies.

AppFlow Policy Binding

To put a policy into effect, you must bind it. You can bind a policy globally, so that it applies to all traffic that flows through the NetScaler system, or to a specific virtual server, so that the policy applies only to the traffic related to that virtual server.

When you bind a policy, you assign it a priority. The priority determines the order in which the policies you define are evaluated. With a NetScaler system, policy priorities work in reverse order. The higher the number, the lower the priority. For example, if you have three policies with priorities of 10, 100, and 1000, the policy assigned a priority of 10 is performed first, then the policy assigned a priority of 100, and finally the policy assigned a priority of 1000.

To avoid complications when adding other policies and to ensure that the policies are evaluated in the order you want, try setting priorities with intervals of 50 or 100 between each policy when you globally bind it. You can then add additional policies at any time without having to change the priority of an existing policy.

When using the configuration utility to bind a policy globally, go to **Configuration** > **System** > **AppFlow** > **Policy Manager**. Using the command-line interface, enter

```
bind appflow global policyName priority gotoPriorityExpression -
type type -invoke (labelType labelName)
```

If you are binding a policy to a specific virtual server using the configuration utility, go to **Configuration** > **Load Balancing** > **Virtual Servers**. Using the command-line interface, enter

```
bind lb vserver name -policyname policy_name -priority priority
```

For additional information about binding policies to the NetScaler system, see the Citrix article CTX132359 at *http://support.citrix.com*.

AppFlow for Virtual Servers

If you want to monitor only the traffic through certain virtual servers, you can enable AppFlow specifically for those virtual servers. You can enable AppFlow for load-balancing, content-switching, cache-redirection, SSL VPN, GSLB, and authentication virtual servers.

To enable AppFlow for virtual servers using the configuration utility, go to the navigation pane and expand the feature node for which you want to enable AppFlow, and then click **Virtual Servers**. For example, if you want to enable AppFlow for a content-switching virtual server, expand **Configuration** > **Content Switching** > **Virtual Servers** and then click **Add**.

When using the command-line interface, enter:

```
set feature_name  -vserver vServerName protocol IPAddress port -
appflowLog ENABLED
```

 © Copyright 2012 Citrix Systems, Inc.

AppFlow Collectors and EdgeSight Monitoring

The EdgeSight Monitoring application provides web page monitoring data with which you can monitor the performance of various web applications served in a NetScaler environment. You can now export this data to AppFlow collectors to get an in-depth analysis of the web page applications. AppFlow, which is based on IPFIX standard, provides more specific information about web application performance than does EdgeSight Monitoring alone.

You can configure both load-balancing and content-switching virtual servers to export EdgeSight Monitoring data to AppFlow collectors. Before configuring a virtual server for AppFlow export, associate an AppFlow action with the EdgeSight Monitoring responder policy. The following web page performance data is exported to AppFlow:

Page load time	Elapsed time, in milliseconds, from when the browser starts to receive the first byte of a response until the user starts to interact with the page. At this stage, all the page content might not be loaded.
Page render time	Elapsed time, in milliseconds, from when the browser receives the first byte of response until either all page content has been rendered or the page load action has timed out.
Time spent on the page	Time spent by users on a page. Represents the period of time from one page request to the next one.

AppFlow transmits the performance data by using the IPFIX format. The AppFlow templates use the following Enterprise-specific Information Elements (EIEs) to export the information:

Client load end time	Time at which the browser receives the last byte of a response to load all the objects of the page, such as images, scripts, and stylesheets.
Client load start time	Time at which the browser receives the first byte of the response to load any objects of the page, such as images, scripts, and stylesheets.
Client render end time	Time at which the browser finishes rendering the entire page, including the embedded objects.

© Copyright 2012 Citrix Systems, Inc.

Client render start time	Time at which the browser starts rendering the page.

Prerequisites for Exporting Performance Data to AppFlow Collectors

Before associating the AppFlow action with the AppFlow policy, verify that the following prerequisites have been met:

- The AppFlow feature has been enabled and configured.
- The Responder feature has been enabled.
- EdgeSight Monitoring has been enabled.
- EdgeSight Monitoring has been enabled on the load-balancing or content-switching virtual servers bound to the services of applications for which you want to collect the performance data.

 © Copyright 2012 Citrix Systems, Inc.

AppFlow Actions and EdgeSight Monitoring Responder Policies

To export the web page performance data to the AppFlow collector, you must associate an AppFlow action with the EdgeSight Monitoring responder policy. An AppFlow action specifies which set of collectors receive the traffic.

When using the configuration utility, go to the navigation pane and expand **Configuration** > **Responder** > **Policies**. In the details pane, select an EdgeSight Monitoring responder policy, and then click **Open**. Then select the AppFlow action associated with the collectors to which you want to send the web page performance data.

Using the command-line interface, enter:

```
set responder policy policyName appflowAction action_Name
```

Third-Party Collectors

When using AppFlow, you can also use third-party collectors such as Splunk and Solar Winds to aggregate the collected traffic information in real-time.

By using both the SolarWinds AppFlow Analyzer Free Tool and AppFlow, you can gain visibility and deep insight into critical business applications. The SolarWinds AppFlow Traffic Analyzer enables you to capture AppFlow records generated from the NetScaler system and to convert those records into easy-to-interpret charts and tables and actionable business intelligence. Together, the AppFlow Traffic Analyzer and NetScaler make it easy to gain insight into customer profiles and trends, optimize application and networking infrastructure, and troubleshoot application issues and threats.

The Splunk for NetScaler App with AppFlow will accept the binary AppFlow data and translate it into time-stamped ASCII text. Splunk will also consume the syslog data produced by the device reporting protocol usage, web application firewall-policy data, load-balancer performance, and VPN data. This data fuels real-time dashboards and reports that allow the NetScaler with AppFlow user to visualize metrics, baselines, and trends that can make transactions more efficient and provide a better customer experience. Once NetScaler and AppFlow data are in Splunk, transaction times can be monitored and a baseline metric can be set over time. Email alerts can be set up to flag abnormal TCP connection times and, if desired, Splunk can run a script to terminate or reset these connections. Splunk can also collect multiline application data and operating system data, which can add additional depth when troubleshooting web connection protocols and server-side applications.

 © Copyright 2012 Citrix Systems, Inc.

Test Your Knowledge

1. Which monitoring method provides visibility at the transaction level for HTTP, SSL, TCP, and SSL_TCP flows?
    a. SNMP
    b. AppFlow
    c. EdgeSight Monitoring
    d. Action Analytics
2. In order for an AppFlow policy to take effect, you must _____ it. (Choose the correct answer.)
    a. bind
    b. assign
    c. monitor
    d. load
3. With what must you associate an AppFlow action in order to export web page performance data to an AppFlow collector?
    a. SNMP trap
    b. AppFlow policy
    c. Content switching
    d. EdgeSight Monitoring responder policy

© Copyright 2012 Citrix Systems, Inc.

© Copyright 2012 Citrix Systems, Inc.

Module 14

Troubleshooting

© Copyright 2012 Citrix Systems, Inc.

Overview

When troubleshooting a NetScaler implementation, you should follow a troubleshooting methodology and reference troubleshooting resources such as the Citrix Knowledge Center, product documentation, Citrix Technical Support, collected NetScaler data, and internal troubleshooting logs.

It is also important to be familiar with NetScaler system architecture and processes. Using troubleshooting tools—either those available through NetScaler or third parties—you can determine the source of failure and resolve it.

After completing this module, you will be able to:

- Identify best practices for troubleshooting NetScaler.
- View, interpret, and manage NetScaler-logged and system information.
- Investigate an issue using the correct utility or information source.
- Investigate and resolve network issues related to NetScaler.
- Troubleshoot the initial configuration of NetScaler.
- Troubleshoot traffic-management issues with NetScaler.

© Copyright 2012 Citrix Systems, Inc.

Troubleshooting Methodology

You may apply the Citrix troubleshooting methodology in a variety of different situations. This methodology can be used as a troubleshooting approach regardless of product, product components, or product environment.

The Citrix troubleshooting methodology consists of the following steps:

1. Define the issue.
2. Gather detailed information.
3. Consider possible causes.
4. Create an action plan to address the issue.
5. Implement the action plan.
6. Observe the results of the action plan.
7. Document changes following issue resolution.

As you discover new issues through user reports or monitoring, you begin the process again.

 © Copyright 2012 Citrix Systems, Inc.

Define the Issue

Defining the issue, by analyzing general information to find the likely cause, is the first step in the Citrix troubleshooting methodology. At this stage, information is limited to user reports about their issues.

You must create an issue statement, which is a concise and accurate summary of the known facts. Using an issue statement places the focus on the true issue and prevents you from troubleshooting any other out-of-scope issues.

Gather Detailed Information

Gathering detailed information is the process of reviewing help desk reports from users who are experiencing the same issue and seeking additional clarity.

More detailed information may be available from tools such as NSTRACE, Wireshark, Citrix Command Center, NetScaler SYSLOG Viewer, Citrix AppFlow, and others. This additional information should include data that serves to exclude other possible causes and helps to narrow the scope of the issue.

For example, by looking at specific data you may notice that a user accessed multiple resources or that network latency was high during a specific timeframe.

Verify that no required information is missing before proceeding to the next steps in the troubleshooting methodology. Identify any recent changes or configurations. Issues created by a change made to the environment may not surface immediately.

Additional Resources

Additional resources for gathering data include:

- User details, which may include screen captures
- Documented steps for reproducing an issue
- Network packet traces
- Network topology diagrams and other deployment documentation
- Database status from your database administrator
- Event logs
- System logs
- Web logs
- Troubleshooting logs
- Citrix Tools as a Service (TaaS), available at *http://taas.citrix.com*
- NetScaler Call Home feature. Call Home registers your NetScaler system with the Citrix Technical Support server (TaaS) and monitors the appliance for common error conditions. If

your appliance is successfully registered with the TaaS server, Call Home automatically uploads system-debug data to that server in the event that one of the conditions occurs. Call Home is supported by any NetScaler MPX appliance running release 10 or later. For more information, see the NetScaler 10 Frequently Asked Questions (FAQs) document at *http://edocs.citrix.com*.

Sometimes the cause of an issue is not readily apparent, and researching the symptoms may be required. A key external resource is the online Citrix Knowledge Center, available at *http://support.citrix.com*.

The Citrix Knowledge Center is the official resource for technical information on Citrix products, including hotfixes, security bulletins, troubleshooting guides, documentation, and white papers. The Knowledge Center contains forums through which external sources can be contacted.

Narrow the Scope

Use the following tasks to narrow the scope of the issue:

Identify symptoms	Further define the scope of the issue by identifying, categorizing, and documenting the symptoms of the issue.
Reproduce the issue	Attempt to reproduce the issue to verify that it still exists and to obtain a clear representation of the environment. Attempt logon by means of a standard user account, following the same steps as the user described. For example, issues related to permissions can often be identified when a standard user account is used.
Investigate the time line	Determine when the issue began and how frequently the issue occurs. For example, the issue occurs at 10:00 daily or started occurring immediately after a change was recorded in the logs.
Determine the scope of the issue	Identify the depth and breadth of the issue by determining how far the issue reaches. The boundary is the limit or scope of the issue.
Check baseline information	Review baseline information, including historical data about the environment and routine-usage information. This information can help determine if recent changes made to the environment may be contributing to the issue.

 © Copyright 2012 Citrix Systems, Inc.

Consider Possible Causes

During this phase in the Citrix troubleshooting methodology, the clues gathered from the investigation in the previous steps will help point to possible causes of the issue. Define the scope of the issue and the corresponding boundaries and create a list of possible causes. This list of possible causes assists in troubleshooting the appropriate areas.

Create an Action Plan

Investigate each possible cause for the issue and create an action plan based on this information. The action plan may involve actions such as configuring the user device and investigating settings.

When creating an action plan, it is important to consider how a resolution for one component may cause another issue. Consider each action carefully before implementing the action plan to determine if the proposed repair may cause adverse side effects. Questions to consider include:

- What is the scope of the change?
- Will the change include elements other than NetScaler?
- When will the change occur, and how long will it take?
- Who will be involved in the change?
- Will the change occur in a controlled environment?
- Is there a plan to revert whatever changes are implemented, if needed?
- If the proposed solution is a workaround, or a fixing solution, do any limitations or drawbacks need to be taken into account?
- In what ways does the test environment differ from the production environment?

When creating an action plan, any potential changes to the environment should first be fully tested in a lab environment. For example, if you suspect that a specific configuration is a cause of an issue, you should test potential corrective actions, such as disabling a feature, before implementing it in production.

Once the solution is determined, create an action plan based on this information. The action plan should be based on the change-control procedure and should include rollback steps in the event that the change is not successful. Using command-line interface commands, or the revision history tool in the configuration utility can assist in documenting change control and rollback procedures. In addition, you need to know the expected benefit and immediacy of the change.

Implement the Action Plan

The next step of the Citrix troubleshooting methodology is implementing the action plan, which could be an investigation leading to the identification of additional action items or a full repair of the issue. When implementing action plans, make only one change at a time. In addition, it is best to make changes in small sets when situations require multiple simultaneous changes. This methodology helps to track the process as well as both the successful and failed actions.

© Copyright 2012 Citrix Systems, Inc.

Ideally, a change should be implemented in a test environment. This will allow trial and error without impacting the whole network. The test environment should mimic as closely as possible the actual live environment to minimize the possibility of unforeseen events happening during implementation. After the solution has been satisfactorily tested, it can then be implemented in the network.

Because of potential unforeseen impacts, it is very important to have a thorough monitoring system in place after any solution has been implemented. Such a monitoring solution can include:

- SNMP alarms for notification of unusual system events
- SNMP polling to gather performance data for trending analysis
- Syslogs forwarding to a remote host
- Citrix Command Center

The next step of the methodology is to observe the results of the action plan; this often occurs while implementing the action plan.

Observe the Results of the Action Plan

Observing results consists of monitoring the changes made to an environment during the implementation of an action plan in order to determine the effectiveness of the repairs. More than one repair may be required to resolve the issue. In addition, you must observe each repair separately to ensure that the action does not create other issues. It is important to gather enough information to determine whether a change is effective, even when the action does not solve the issue entirely.

If the repair is successful, move on to the next step of the Citrix troubleshooting methodology. Otherwise, you must gather more information and create another action plan.

Document Changes Following Issue Resolution

Documentation is an integral part of troubleshooting and change control. Recording changes and, in some instances, the time these changes occurred provides valuable information for future reference. Before closing a case, you must document all troubleshooting steps and actions taken. In general, while working through the initial action plan, you should consider that something that is changed during the repair would affect something else that you failed to notice. For this reason, it is important to maintain documentation to record the changes made. Changes made during the repair process can easily be retraced if necessary.

Documentation provides the following advantages:

- It explains exactly what was done to repair the issue, which can be helpful in repairing similar issues.
- It provides valuable historical information to everyone responsible for maintaining and troubleshooting the environment.

 © Copyright 2012 Citrix Systems, Inc.

- It includes troubleshooting steps taken and their results, which can reduce the likelihood of repeated actions and promote faster case closings with escalated issues.

Citrix recommends documenting all actions, including key information provided by the user, as they occur. Doing so ensures that no vital information is overlooked. In addition, documenting while troubleshooting or on a call reduces the time dedicated to documentation after the case is closed.

Troubleshooting Resources

Useful troubleshooting resources include:

- Citrix Knowledge Center
- Citrix product documentation
- Citrix Technical Support
- Collected NetScaler data
- Troubleshooting log
- Citrix Tools as a Service (TaaS)

Citrix Knowledge Center

An important tool for research, the Citrix Knowledge Center is the official resource for technical information on Citrix products, hotfixes, security bulletins, and troubleshooting guides. The Knowledge Center contains forums through which external sources can be contacted. It also contains a thorough library of documentation and white papers. The forum community is also very knowledgeable and responsive.

For more information about the Citrix Knowledge Center, see Citrix articles at *http://support.citrix.com*.

Citrix Product Documentation

The most recent NetScaler product documentation is available at the Citrix eDocs web site, located at *http://edocs.citrix.com*. The product documentation may also be downloaded through the Netscaler graphical user interface.

Citrix Technical Support

Citrix Technical Support can be an extremely useful resource when troubleshooting a NetScaler deployment. When contacting Technical Support, compile all the previously gathered information pertaining to the issue. Be prepared to engage the support engineer in an exchange of ideas.

Collected NetScaler Data

Gathering all available NetScaler data is important when beginning analysis. Data resources include:

- Results of the `show techsupport` command, which provides configuration files, performance log data, system messages and other relevant system information
- User feedback, which may include screen captures

 © Copyright 2012 Citrix Systems, Inc.

- Documented steps for reproducing the issue
- Network packet traces
- Syslogs
- Web logs
- SNMP alarms
- Network topology diagrams and other deployment documentation

Troubleshooting Log

A troubleshooting log should not be confused with the various log files the NetScaler system generates. Troubleshooting logs document any previously encountered issues and their resolutions. Such logs are valuable when researching a current issue.

A troubleshooting log may be a simple spreadsheet that includes information such as:

- Issue type
- Description
- Date entered
- Entered by
- Priority
- Assigned to
- Issue resolution
- Date resolved

© Copyright 2012 Citrix Systems, Inc.

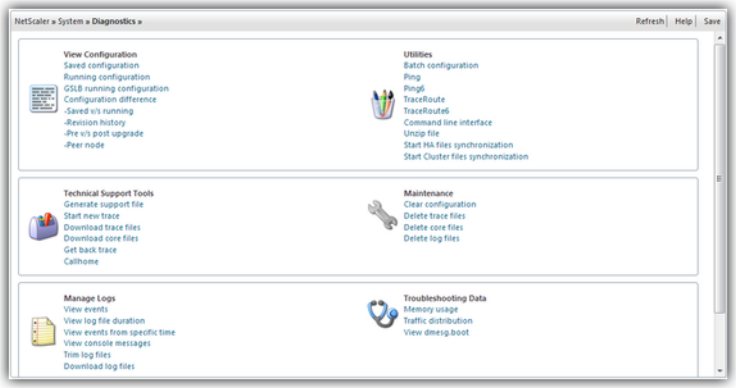
Troubleshooting Tools

NetScaler has a number of tools to help with troubleshooting. These include the following tools that are available through the BSD shell:

- nsconmsg
- nstrace.sh

In addition, the configuration utility contains a diagnostic utility (located in **System** > **Diagnostics**) that provides access to the following tools:

- traceroute
- ping
- tcpdump



nsconmsg

The nsconmsg command displays various NetScaler system statistics. The nsconmsg command collects and displays all the statistics and system information from the NetScaler system. This information can be used to analyze traffic and the NetScaler system.

The nsconmsg command is used for viewing:

- Events
- Console messages
- CPU/memory statistics
- Interface statistics
- Commands executed by users
- Feature-specific statistics

The nsconmsg command polls the NetScaler kernel to extract performance records. Performance records are written to disk through the nsconmsg process. These performance records contain logs

 © Copyright 2012 Citrix Systems, Inc.

and statistics that help with troubleshooting the system or a specific feature. Live or historical statistics can be gathered directly through the nsconmsg command.

The nsconmsg application is governed by the nslog.sh script, which executes at startup time. When this application starts, it connects to the specified NetScaler system and begins to collect the data. It has the ability to log the binary data to the file or to view information online. Filters are defined to extract only the required information. The script also manages log rotation.

> The nsconmsg utility runs in the BSD shell and not the command-line interface.

newnslog

The /var/nslog/newnslog binary file stores console messages, events, and performance statistics. The nsconmsg command displays the contents of the newnslog file, which is updated every seven seconds. Like any automatically created log file, newnslog can be daunting to read. The nsconmsg utility can trim the newnslog to display only requested information.

Archived log files are stored in gzip format. Enter the following command to view them:

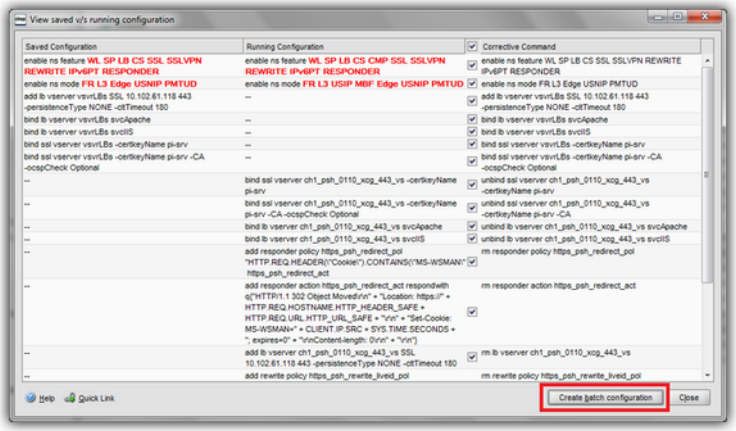
```
zcat filename | nsconmsg -K pipe options
```

Then append options as you would normally with nsconmsg.

> The nsconmsg command is case sensitive. Make sure to differentiate between the -K (input file) and the -k (output file). A common administrative issue is incorrect use of -K for the input file and overwriting the log file. Best practice is to back up the log file and not to modify the original.

Comparative Configuration

To reduce the resolution time on issues, many useful tools and utilities are provided in the NetScaler user interface. These tools are located in the **System** > **Diagnostics** page of the configuration utility. For example, you can compare the configuration files from a previous version of NetScaler for the purpose of reverting to the prior version. The Saved Configuration link displays the configuration from the `ns.conf` file which is saved on the NetScaler system. The Running Configuration displays the configuration which is currently running on NetScaler. The running configuration can display more than the saved configuration because it captures information not yet saved in the `ns.conf`. The configuration utility allows up to five previous versions. The GSLB running configuration is also available.



The Configuration Difference link displays the difference between the saved configuration and the running configuration on the NetScaler. If there is a difference, then the difference is displayed, corrective commands are listed, and a batch file can be created to apply the commands. If there is no difference, then a "Configuration is identical" message is displayed.

The Revision History link compares the running configuration with multiple revisions as saved in the Netscaler configuration. This tool is helpful for troubleshooting because you can determine what has changed since the last revision, or previous revisions, that could be causing an issue.

The Pre vs Post Upgrade link compares the running or saved configuration with the configuration before the upgrade. In addition, the **Peer Node** link can display the difference between primary and secondary devices in a high-availability deployment.

Citrix Command Center can also be used to display a detailed log of changes and backups of configuration files, and allows for up to 50 previous versions.

Show TechSupport Command

When troubleshooting NetScaler, you can generate a set of files required for troubleshooting any issue related to Netscaler. A support file can also be generated from the configuration utility using the Generate support file link in the **System** > **Diagnostics** page under Technical Support Tools.

From the command line, you can also run the following command:

```
show techsupport
```

 © Copyright 2012 Citrix Systems, Inc.

Using the Visualizer

The network visualizer is a tool that you can use to view the network configuration of a Netscaler node, including the network configuration of the nodes in a high availability deployment. You can also modify the configuration of VLANs, interfaces, channels, and bridge groups, and perform high availability configuration tasks.

The visualizer can be a helpful tool for troubleshooting NetScaler. For example, when you are logged on to a standalone system, you can use the visualizer to view a consolidated graphical summary of key network components. You can also view the individual details of various network components, as well as view node details, node statistics, and statistics for VLANs and interfaces. These capabilities can potentially help you find the source of a problem or issue in your environment.

For more information about the visualizer, see Citrix eDocs at *http://edocs.citrix.com*.

Test Your Knowledge

1. Which of the following tasks does NOT help narrow the scope of the issue during troubleshooting?
    a. Identify symptoms
    b. Reproduce the issue
    c. Investigate the timeline
    d. Create a list of all issues
2. Where is the most recent NetScaler product documentation located?
    a. www.citrix.com
    b. www.google.com
    c. http://edocs.citrix.com
    d. www.citrix.com/netscaler
3. Which function does the `show techsupport` command perform?
    a. Contacts Citrix Technical Support
    b. Displays a list of potential issues in your environment.
    c. Displays a history of interactions with Citrix Technical Support
    d. Generates a set of files required for troubleshooting an issue with NetScaler.

 © Copyright 2012 Citrix Systems, Inc.

Networking

Networking issues include:

- Duplex mismatch or misconfigured interface settings
- Unresponsive system
- Inaccessible content

Several common issues can be checked and discarded early in the troubleshooting process, including a slow NetScaler system due to a duplex mismatch, an unresponsive system, and inaccessible content.

> The NetScaler system always draws power when it is plugged in. Therefore, an occasional simple restart does not clear severe console failure response conditions. Completely remove the power from the units (unplug them from the outlets) for 30 seconds, and then turn them back on.

Duplex Mismatch or Misconfigured Interface Settings

If the duplex setting on the NetScaler system does not match the setting on the switch or router on which it is connected, the NetScaler system may be slow to respond to requests. To investigate this issue, click **View console messages** in the diagnostics pane of the system node.

IP address conflicts or a duplex mismatch alert should be checked. Interfaces can be configured in the interfaces screen of the network node.

Enter the following command in the command-line interface to correctly set the interface:

```
set interface id -speed speed -duplex duplex_mode
```

The following table lists available arguments:

Argument	Description
`id`	Specifies the interface ID, for example 1/1 or 1/5
-speed `speed`	Specifies the network speed, which can be set to Auto, 10, 100, or 1000
-duplex `duplex_mode`	Specifies the duplex mode, which can be Auto, Full, or Half
n/n	Specifies the appropriate interface

Enter the following command to obtain detailed interface statistics and switch port settings:

```
sh interface n/n
```

> Incorrect interface settings usually result in an interface that will not display. It is important to ensure that the configuration on the NetScaler interface closely resembles the configuration on the corresponding switch/router interface. Ensure that speed, duplex, and flow control settings match.

Inaccessible Content

If content located behind the NetScaler system is inaccessible, the following items should be verified:

- Have configuration changes been made to servers or network devices?
- Have configuration changes been made to server, service, or virtual server objects?
- Can the site be accessed directly (for example, bypassing the NetScaler system)?
- Can the server and port be accessed using Telnet? Firewall port configuration is a common cause of inaccessible systems.

Testing of Network Connectivity Through a Firewall

You can test network connectivity from the NetScaler through a firewall to the Internet or intranet. For example, for intranet connections you can configure a service object referencing the server or port you are trying to connect to and test the connection. This causes the NetScaler to attempt a TCP connection on the specified port, with the traffic being sent from the SNIP.

Using telnet is not the best method for this situation, because it uses the NSIP as the source IP, and production traffic will be sent from the SNIP. If the service does not start, then it is likely that firewall or routing issues exist.

 © Copyright 2012 Citrix Systems, Inc.

Display NetScaler System Information

The model and serial numbers of a NetScaler system can be useful during troubleshooting.

To display the model and serial numbers, run the following command from the command-line interface of the system:

```
show hardware
```

This information can also be viewed in the system node of the configuration utility.

Troubleshooting Hardware Issues

Examples of hardware issues include:

- The interface is not detected
- The interface is always flapping, or alternately showing as UP and DOWN
- Continuous console messages referring to a hardware component
- Lights do not appear on the appliance when it is turned on (assuming you have already tested your power sources and cables)
- Network interface LED lights do not appear when a network cable is inserted
- Cannot connect to the console or get a logon prompt through a serial port (assuming you use the correct terminal emulation settings explained in the "Configuring the Application Switch Using the CLI" section of Citrix article CTX112893 at *http://support.citrix.com*.
- BIOS or POST errors
- Specific SSL card failures
- Output of the `dmesg` shell command indicates interface issues or read/write errors on the hard disk sector
-

> Disk errors may result from file corruption, which can be corrected by running a file consistency check. Citrix Technical Support can help you execute these procedures.

- You received an appliance with copper interfaces instead of fiber, or vice versa

Additional data required to troubleshoot hardware issues include:

- The "newnslog" file
- The "dmesg" shell command output
- The "show node" output from the NetScaler command-line interface if the issue is related to SSL card failures

In some cases, it may be possible to recover the unit from errors. Contact Citrix Technical Support if you are unable to resolve hardware issues and to obtain a Return Materials Authorization (RMA). For more information about NetScaler hardware issues, see Citrix articles CTX113462 and

CTX109304 at *http://support.citrix.com*. In addition, refer to the Citrix Technical Support Brief Troubleshooting Guide at *http://support.citrix.com*.

Non-Maskable Interrupt (NMI) Button

If the NetScaler is not responding, you can force a core dump and restart the physical appliance, using the NMI button. The core files can help you or Citrix Technical Support investigate why the NetScaler is not responding. The NMI button is the recessed red button on the back side of a physical NetScaler appliance.

The NMI button does not exist on a NetScaler VPX.

When you press this button, the appliances dumps troubleshooting data and restarts. The vmcore and kernel files are then available in the `/var/crash` directory. The process of dumping the data and restarting the appliance can take between 10 and 45 minutes, depending on the RAM of the appliance. For more information about the NMI button, see Citrix article CTX120660 at *http://support.citrix.com*.

Display Software Information

NetScaler systems are pre-loaded with software from the factory. When troubleshooting software issues, the following commands can be used to display the current software information, such as the version, feature, and license information:

```
show feature
show version
show license
```

Software information is also available in the configuration utility. The version is displayed in the top pane, and the license information is located in the **System** > **License** menu.

 © Copyright 2012 Citrix Systems, Inc.

Troubleshooting License Issues

You can view license issues that have been read by entering the following shell command:

```
cat /var/log/license.log
```

If a feature that you need licensed is listed, you must request a new license. If "YES" is listed next to a feature, the feature can be enabled by right-clicking the feature in the configuration utility.

An example of a licensing issue is an error message displaying a restriction such as "Certificate with key size greater than 512 bits not supported." This message displays because a license is not applied or a public NetScaler software release is not installed on the system. To resolve the issue, you can apply the appropriate license to the NetScaler system, or upgrade the software of the system to a public NetScaler software release, or do both. For more information about troubleshooting license issues, see Citrix article CTX125548 at *http://support.citrix.com*.

Load-Balancing Issues

Load-balancing issues with NetScaler include either services or virtual IP (VIP) flapping, uneven load balancing, and slow responses when accessing applications through VIP.

The newnslog file, the URL and logon credentials for a site (if public), and the following commands are useful data for troubleshooting load-balancing issues:

```
show lb vserver <vservername>
```

```
show service <servicename>
```

A trace is useful for problems related to packet flow or handling. Information about running a packet trace is provided in the NSTRACE topics within this module.

The following command, entered from the shell console, will display load-balancing information:

```
ns trace nsconmsg -K /var/nslog/newnslog -s ConLb=1 -d oldconmsg
```

Uneven Load Balancing

The following table lists items to check when trying to explain and diagnose uneven load balancing:

 © Copyright 2012 Citrix Systems, Inc.

Item	Description
Slow start	The NetScaler system performs a slow start to avoid overloading physical servers. During the slow start phase, the NetScaler system distributes requests by round robin, regardless of the actual load-balancing method configured on the virtual server. However, it honors the configured weight on bindings even during round robin. A slow start occurs in any of the following conditions: • The load-balancing method changes. • A new server is bound to a virtual server. • An existing server binding is removed from a virtual server. • A server changes its status from DOWN to UP. A contact slow start indicates service flapping, or alternating between showing as DOWN and UP. For more information about the slow start mode, see Citrix article CTX124711 at *http://support.citrix.com*.
Persistence	When enabled, persistence may create uneven loading because the NetScaler system must direct requests from a specific client to a specific server. Only unique or expired clients get balanced according to the load-balancing method.
Inconsistent server performance	Performance across a set of servers is seldom consistent. Therefore, some servers serve requests faster and, based on the load-balancing method configured, will likely cause an uneven balance. The degree of imbalance increases as the degree of inconsistency between the servers increases.
Service weights	Service weights can result in uneven load balancing. Lower-weighted services serve fewer requests.

© Copyright 2012 Citrix Systems, Inc.

Service or Virtual Server Flapping

Flapping in this situation occurs when a service or virtual server alternates between showing as UP and DOWN. A service most likely flaps because its monitors are failing.

Enter the following command in the command-line interface to obtain detailed monitoring and status information:

```
show service service name
```

In most cases, assume that the monitor is failing legitimately and troubleshoot the issue on the servers themselves. It is also possible that the monitor configuration is causing service flapping by monitoring too frequently or for the wrong response. The configuration should then be inspected for any issues.

A virtual server goes DOWN when all of the bound services go DOWN, so a virtual server that flaps is likely the result of services that are flapping.

Load Balancing of Third-Party Servers

Many NetScaler implementations involve the deployment of NetScaler with a third-party server such as Microsoft, SAP, Oracle, or others. On the Citrix Development Network and online Citrix Community, you can access NetScaler deployment guides for Microsoft Lync Server, Exchange, SharePoint, Office Communication Server, and others. The guides are located at *http://community.citrix.com/display/ns/Microsoft*.

In addition, load-balancing content and deployment guides for SAP are located at *http://community.citrix.com/display/ns/SAP* and a whitepaper on using NetScaler to scale and optimize Oracle MySQL is located at *http://www.citrix.com*.

 © Copyright 2012 Citrix Systems, Inc.

Authentication, Authorization, and Access (AAA) Issues

Authentication, authorization, and access (AAA) issues may cause content located behind a NetScaler system to become inaccessible. First, answer the following questions:

- Have configuration changes been made to servers or network devices?
- Have configuration changes been made to server, service, or virtual server objects?
- Can the site be accessed directly (in other words, bypassing the NetScaler system)?
- Can the server and port be accessed using Telnet?

It is also important to gather data before troubleshooting. Useful commands include the following:

```
stat aaa (displays aaa statistics)
```

```
show aaa (display information for users, groups, parameters)
```

For more information about troubleshooting, see Citrix articles CTX133789, CTX131684, and CTX120639 at *http://support.citrix.com*.

Authentication Troubleshooting

When troubleshooting LDAP, or any type of authentication with Citrix NetScaler, a useful tool is the Aaad.debug module. Using this tool, you can troubleshoot authentication issues such as general authentication errors, username or password failures, authentication policy configuration errors, and group extraction discrepancies.

The Aaad.debug tool displays information in realtime, including group extraction and authentication failures, which can be very useful for determining which authentication provider is causing the issue.

To use the Aaad.debug tool, begin at the command line, access the shell, change to the `/tmp` directory, and begin the debugging process by entering the following command:

```
cat aaad.debug
```

For more information about troubleshooting the authentication process, see Citrix article CTX114999 at *http://support.citrix.com*.

Another useful way to troubleshoot authentication is by running a network packet trace if it is a network problem.

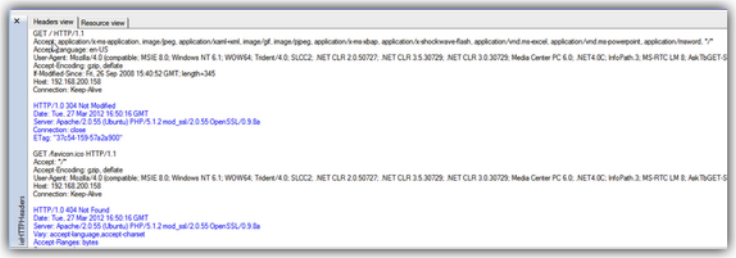
For more information about troubleshooting Active Directory group extraction for LDAP, see Citrix article CTX125797 at *http://support.citrix.com*.

© Copyright 2012 Citrix Systems, Inc.

SSL Troubleshooting

SSL issues involve client and server network traffic, so viewing HTTP headers when troubleshooting these issues is useful. The following free tools are available for this task and can be found by performing a web search:

- Live HTTP Headers for Mozilla Firefox
- IE HTTP Headers for Microsoft Internet Explorer
- HTTP Headers Extension for Google Chrome



Header review is a well-known and effective method of troubleshooting connections for persistence failures when using cookie persistence, isolating authentication issues, and back-end server response issues.

These tools provide you with the client view of the connection as it happens. As with other HTTP connections, you are able to see the server responses but will be missing the SSL details as everything is decrypted for the client before the headers are displayed. Therefore, missing portions of the SSL handshake and the certificate exchange are not displayed. These tools also allow you to save the headers for future review or sharing with others.

The Google Chrome Extension also includes the option to get client-side packet captures.

SSL Command-line Interface Commands

When you use a show command, you typically extract a snapshot of configuration data and performance information from the NetScaler kernel. However, many commands show additional data if an exact object is selected. For example, the following command displays more specific information when you specify a virtual server:

```
show ssl vserver vserver name
```

More common commands include the following:

```
show ssl certkey
```

 © Copyright 2012 Citrix Systems, Inc.

```
show ssl stats (stat ssl)
```

```
show ssl service service name
```

```
show ssl certlink
```

If you suspect a global setting may be related to, or the cause of, your issue, you can list these at the command-line interface with the following command:

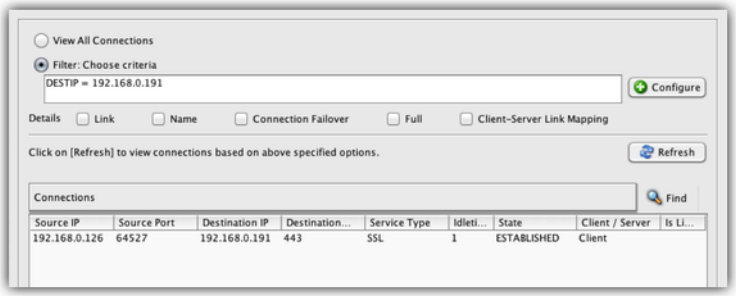
```
show ssl paramter
```

This command will display advanced global ssl parameters. It can be beneficial to check these global setttings to determine if one of them is the cause of the issue.

Troubleshooting of Encrypted SSL Connections

With encrypted SSL connections, since the packet data is encrypted there are just a few options for troubleshooting. TCP layer and below-the-TCP layer information is viewable in the nstcpdump and nstrace files. You can also verify connections by reviewing the connection table on the NetScaler system.

In the configuration utility, this information is located in the **System** > **Diagnostics** menu. Then, click **Monitor Connections** > **TCP/IP Connections** . You can view all connections or filter by IP, port, service type, and other options.



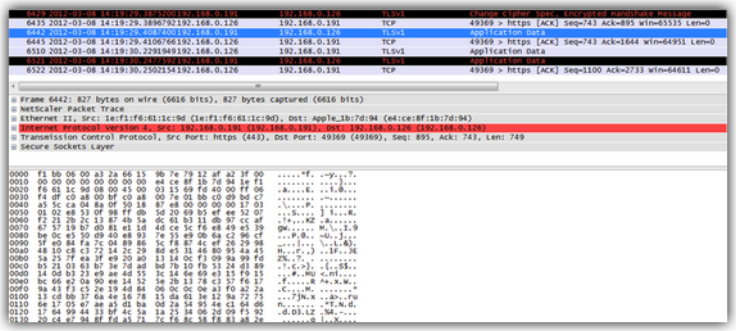
Other options for troubleshooting encrypted SSL connections include packet level analysis and a Wireshark capture. Nstcpdump is a standard tcpdump with a wrapper around it to allow for some NetScaler-specific information, and nstrace is a tool used for creating capture files for deeper analysis. Technical support will often ask for nstrace captures, because it adds information to the trace that the nstcpdump output will not provide, and provides you with the syntax to use for each case. Tcpdumps are widely used for troubleshooting, because they provide live data on the layer-3 connectivity, but is still limited to ensuring that the TCP connection is working or not.

A protocol analyzer such as Wireshark can help you view the packet data to see where in the connection the issue is located. For more information about using Wireshark with Citrix Netscaler, see Citrix article CTX122318 at *http://support.citrix.com*.

© Copyright 2012 Citrix Systems, Inc.

Decoding SSL Traffic with Wireshark

Wireshark is a third-party tool that can be used to analzye SSL traffic. For example, you can use the NetScaler packet trace tool to create capture files, then use Wireshark to expose and view the header.



Decoding traffic with Wireshark requires a few key pieces of information which should be easily accessible after you have run a packet trace and downloaded it to your workstation:

- SSL vServer IP address where the client is connecting
- Port number
- Key file and the password for the key, if necessary

Tips for executing a successful capture that will be decrypted or decoded include the following:

- On the SSL vServer, ensure that SSL session reuse is disabled. From the command line, run the following command:

  ```
  set ssl vs test -sessReuse DISABLED -sessTimeout 120
  ```

- Capture the entire handshake. Citrix recommends that you close your browser. Start the trace on the NetScaler system, then start the client connection process. This will ensure that the entire handshake is in the file and available for capture.
- Remove the password protection from the key files, if necessary. You can use the OpenSSL third-party tool to remove the password. For more information about using OpenSSL, see *http://www.openssl.org/source*.
- If you have exported the key from a web server, you may need to extract the key using OpenSSL. The resulting output file can be read by a text editor and copied into its own file for troubleshooting purposes.

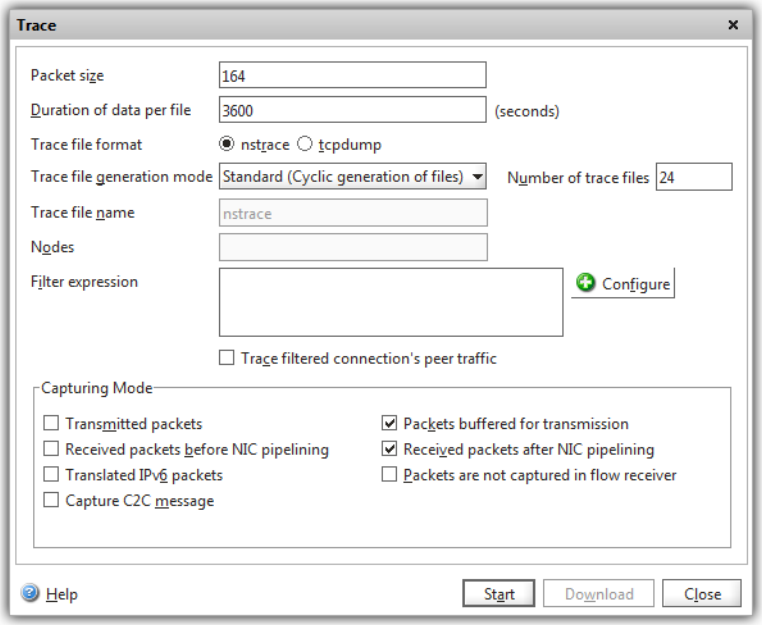
For more information about using Wireshark, see *http://www.wireshark.org*.

 © Copyright 2012 Citrix Systems, Inc.

Network Traffic Capture using NSTRACE.SH

Network traffic can be captured using the nstrace.sh script and the nstcpdump.sh script on the NetScaler system.

The nstrace.sh script is used for troubleshooting and provides extended data for analysis. Filters can be set on the nstrace.sh script to capture a trace with the front-end (client to virtual server) or the backend (NetScaler system to server), or both. To limit the file size, the nstrace.sh script should run for the minimum amount of time needed to gather the appropriate information.



The nstrace.sh script provides:

- Packet capture in the native NetScaler trace format, which provides NIC device information and packet-transmission data
- Connection link information, which allows for the identification of links between client and virtual server, and SNIP/MIP-to-server TCP connections
- The ability to create multiple files based on a set amount of time and number of files for each cycle using the -nf and -time options
- The ability to create separate trace files for each NIC using the -nic option
- Packet capture in tcpdump format, if desired

© Copyright 2012 Citrix Systems, Inc.

Network Traffic Capture Using NSTCPDUMP

The nstcpdump.sh script, run from the shell utility, is used for low-level troubleshooting, or troubleshooting when low-traffic volume does not support the nstrace.sh script. The nstcpdump.sh script does not collect all the information that the nstrace.sh script can collect; this point should be considered before using the nstcpdump.sh script for troubleshooting.

The nstcpdump.sh script should not be used if a complete output of the nstrace.sh script is needed for the appropriate analysis of a NetScaler system issue.

Filters based on IP address, TCP, or UDP port numbers can be set for the output of the nstcpdump.sh script. This output provides targeted data which can more easily be analyzed.

NSTCPDUMP syntax

The nstcpdump.sh script is run at the shell prompt by using the following syntax:

```
nstcpdump.sh [options] [filter_expression]
```

If a packet count is not specified in the syntax, the nstcpdump.sh script is stopped by running the kill command. For more information about nstcpdump.sh, see Citrix article CTX118185 at *http://support.citrix.com*.

TCPDUMP Options

The nstcpdump.sh script supports the use of tcpdump options to allow you to capture specific data.

-c	Specifies the number of packets to record and to automatically terminate the script.
-X	Sends the script output to stdout, and displays packet content in hexadecimal and ASCII format.
-w <destination_file>	Writes the recorded packets to the specified file in standard tcpdump format. The complete path is used when specifying the file.
-I <interface_number>	Restricts packet recordings to a specific interface. This option is used multiple times to select multiple interfaces.

 © Copyright 2012 Citrix Systems, Inc.

NSTRACE Syntax and Options

The nstrace.sh script is run from the BSD shell using the following syntax:

```
nstrace -sz postiveInteger -tcpdump  ENABLED | DISABLED
```

Replace `postiveInteger` with a value between 0 and 100 to specify the file size, where 0=unlimited and 100=maximum file size.

For more information about capturing an trace from the command line, see Citrix article CTX120941 at *http://support.citrix.com*.

The nstrace.sh script supports the use of options to allow you to capture specific data.

-mode [mode]	Identifies the capturing mode for the trace. Mode values include: TX - transmitted packets; TXB - packets buffered for transmission; RX - received packets before NIC pipelining; IPV6 - translated IPv6 packets; and NEW_RX - received packets after NIC pipelining.
-name	Specifies the name of the trace file
-filter	Specifies the filter expression for trace filtering
-id	Specifies a unique ID for the trace file name and should only be used with the -name filter

Troubleshooting with Filter Expressions

The start new trace utility in the diagnostics menu of the configuration utility helps collect the live trace from the NetScaler system. Many connection level or latency-related issues require a trace file for troubleshooting. The start new trace utility helps you collect the correct trace file, if you choose the correct options. One of the options to configure is the filter expression field.

Filter expressions are important because they allow you to target the core problem area. You can add expressions to perform the filtering.

The expression qualifiers could be any of the following:

- Source IP
- Source port
- Destination IP
- IP address
- Port
- Service type
- Idle time
- State
- Service name
- Virtual server name
- Connection ID
- Interface
- VLAN

The filter contains many options for customization, including tracing filtered-connection peer traffic. This is useful when you specify only a source IP as a filter but also want to see the server-side connection for source-IP-originated client connections.

 © Copyright 2012 Citrix Systems, Inc.

SSL Offload Troubleshooting

SSL offloading issues include:

- Access to the SSL VIP address failing
- Certificate-related warnings occurring
- Intermediate certificate improperly linked
- Browser warning showing a web page that is not secure

Access to SSL VIP Address Failing

This issue typically occurs when the certkey (certificate-key pair entity on the NetScaler system) is not bound. If the certkey is not bound, the status of the virtual server will display as DOWN.

Enter the following command in the command-line interface to show the state:

```
show ssl vserver vserver_name
```

The following table lists the available arguments:

Argument	Description
`vserver_name`	Specifies the name of the SSL virtual server

Enter the following command in the command-line interface to resolve the issue:

```
bind ssl certKey vserver_nameservice_name certkey_name
```

The following table lists available arguments:

Argument	Description
`vserver_name`	Specifies the name of the SSL virtual server to which the certificate-key pair needs to be bound
`service_name`	Specifies the name of the SSL service to which the certificate-key pair needs to be bound. Use the `add service` command to create this service.
`certkey_name`	Specifies the object name for the certificate-key pair

© Copyright 2012 Citrix Systems, Inc.

Intermediate Certificate Not Linked

Certain server certificates may be issued by default by a CA that is not in the browser's trusted store. In this case, an intermediate certificate, which establishes the chain of trust, needs to be added to the NetScaler system and linked to the certkey. When the certificate is presented to the end user, the intermediate certificate is also provided.

 © Copyright 2012 Citrix Systems, Inc.

Troubleshooting Policies

The need to troubleshoot policies may arise because a configured policy does not function properly in your environment, or because of issues encountered while evaluating policies. Using specific parameters, or the failure to set a specific parameter, can cause an issue with a policy. It is also helpful to verify that the policy was applied to the correct server.

For example, you may need to use a specific operator for a parameter in an advanced policy to ensure that the NetScaler identifies UTF-8-encoded characters existing in a URL. For more information about UTF-8 encoded characters, see Citrix article CTX125766 at *http://support.citrix.com*.

It is helpful to assess all policies bound either globally or at the virtual server level to determine which policies are applied and what could be causing an issue.

Access control lists or responder policies can be used to terminate traffic from a specific IP address when a termination policy is not functioning and the web pages are still displayed. For more information about content filtering policies not terminating traffic, see Citrix article CTX124081 at *http://support.citrix.com*.

Caching

The following table lists cache issues:

Issue	Description
Incorrect content cached	Improper configuration of the Integrated Caching feature may cause users to see incorrect content served from the cache. Enter the following command in the command-line interface to determine whether a particular object is in the cache: `show cache object -url url -host hostname` Enter this command after making an initial request for the object (for example, using a web browser) to give the NetScaler system a chance to cache the object. If the object is not in the cache, the NetScaler system reports "ERROR: No such resource." To determine if an object served from the cache does not match the object served from the back-end web server(s), compare the HTTP responses from: • A client with a connection through the NetScaler system • A client with a connection not through the NetScaler system If the responses differ, the responses received from clients passing through the NetScaler system are erroneous and the issue requires troubleshooting. Verify if there are any cache selectors configured that could serve the same object for different VIP addresses.
Expired content served	Proper expiry headers are not being provided; check the content on the servers and the invalidation parameters.

 © Copyright 2012 Citrix Systems, Inc.

Issue	Description
Cache expiry causes traffic surge to the back-end	Decrease the timeouts for the cached content or create different content groups that expire at different times, so all the content is not expiring at the same time. Enable the prefetch option.

By default, the global caching limit is set to 0; therefore, no caching occurs. It is necessary to set this value and restart the NetScaler for caching to occur. For more information about caching, see Citrix article CTX124553 at *http://support.citrix.com*.

Use of Telnet or Monitors to Check Ports

NetScaler uses a FreeBSD shell architecture that has a number of built-in tools. These shell utilities can be used to test traffic flows and for troubleshooting. One of these tools is telnet, which helps troubleshoot connectivity issues between the NetScaler system and other devices on a specific port. For example, if you are experiencing connectivity issues, sending the telnet command to the correct port should verify that nothing is blocking the communication.

Another way to check port connectivity is to create a service for the problematic component using the required monitor. Traffic will then use the SNIP or MIP as the source address, rather than the NSIP. This will assist in confirming communication between the NetScaler and the service.

Telnet can also be used to troubleshoot an SMTP connection to a server. For more information about the use of Telnet or monitors to check ports, see Citrix article CTX129869 at *http://support.citrix.com*.

 © Copyright 2012 Citrix Systems, Inc.

High Availability

High-availability issues include:

- Configuration synchronization failure
- File synchronization failure
- Unexpected failover

Configuration Synchronization Failure

Synchronization failure can be a result of connectivity issues, duplex mismatches, packet drops, or the /netscaler/nsnetsvc process not running.

Perform the following tasks if synchronization between the primary and secondary node fails:

- Verify that the primary and secondary nodes can communicate with each other. Management and heartbeat messages are sent through layer-2 protocols. Layer-2 connectivity between the two high-availability nodes must allow the heartbeat to be received within 3 seconds.

> One node will display a status of listening in this scenario.

- Ensure that any configured ACLs permit communication between the pair.
- Enter the following command to check the `inetd.conf` file to ensure that the /netscaler/nsnetsvc process is not disabled:

```
ns# more /etc/inetd.conf
```

- Ensure that the `nsnetsvc stream tcp nowait root /netscaler/nsnetsvc nsnetsvc` line contains comments.
- Enter the following command in the shell to check the `ns_com_cfg.conf` file on the secondary node:

```
ls -l
```

- Ensure that the /tmp directory has write permissions. For example:

```
drwxrwxrwt  4 root  wheel  512 Aug 17 21:28 /tmp
```

- Verify that the two nodes are not running different versions of the NetScaler operating system.

File Synchronization Failure

Both nodes in a high-availability pair may need a set of common-configuration or certificate files, depending on the deployment needs. If so, files may need to be manually synchronized. For

example, if SSL offload is enabled, then SSL certificates must be copied to the same location (directory) on both nodes. Similar examples include vsr.html (for SureConnect), any manually customized files, or any other batch files containing configuration commands.

Enter the following command in the command-line interface to manually synchronize files between nodes in a high-availability pair:

```
sync ha files mode
```

The following table lists available arguments:

Argument	Description
`mode`	Specifies the sync mode. Possible values include: • all • bookmarks • ssl • htmlinjection • imports

The following table lists paths corresponding to synchronization mode:

Mode	Path
all	/nsconfig/ssl/ /var/vpn/bookmarks/ /nsconfig/htmlinjection/
ssl	/nsconfig/ssl/
bookmarks	/var/vpn/bookmarks/
htmlinjection	/nsconfig/htmlinjection/

Unexpected Failover

If the NetScaler systems are failing over unexpectedly, enter the following command in the command-line interface to view current events that may be causing the failover.

```
root@ns# nsconmsg -d event
```

Possible causes include:

 © Copyright 2012 Citrix Systems, Inc.

- An interface is down
- An SSL acceleration card is down
- The primary node has crashed

Test Your Knowledge: Troubleshooting

1. Which of the following are NOT considered networking issues?
    a. Inaccessible content
    b. Unresponsive content
    c. BIOS or POST errors
    d. Duplex mismatch or misconfigured interface settings
2. Which of the following are useful data for troubleshooting load-balancing issues? (Choose two.)
    a. newnslog
    b. license.log file
    c. Administrator credentials
    d. `show lb vserver` command
3. Which of the following is NOT considered a high availability issue?
    a. Unexpected failure
    b. Service or virtual server flapping
    c. File synchronization failure
    d. Configuration synchronization failure

 © Copyright 2012 Citrix Systems, Inc.

© Copyright 2012 Citrix Systems, Inc.

851 West Cypress Creek Road Fort Lauderdale, FL 33309 USA (954) 267 3000 www.citrix.com

Rheinweg 9 8200 Schaffhausen Switzerland +41 (0) 52 63577 00 www.citrix.com

© Copyright 2012 Citrix Systems, Inc. All rights reserved.